

(1) HTML

Syntax

< starting Tag > Hypertext < / ending >

(1) Text tag

① Heading Tag - ⑥

< h₁ > AB < / h₁ >

< h₂ > AB < / h₂ >

< h₃ > AB < / h₃ >

< h₄ > AB < / h₄ >

< h₅ > AB < / h₅ >

< h₆ > AB < / h₆ >

② Paragraph Tag -

< P > AB < / P >

③ Pre Tag

< Pre > AB < / Pre >

② Structure Tag

- ① HTML tag `<html>`
- ② Head tag `<head>`
`</head>`
- ③ Body tag `<body>`
`</body>`
`</html>`

③ Text formatting tag

- ① underline \rightarrow `<u> AB </u>`
- ② Italic \rightarrow `<i> AB </i>`
- ③ Bold tag \rightarrow ` AB `

④ Meta tag

meta data

Data about data

`<title> Demo html </title>`

Extension of html file is written as .html

heading size decreases from h_1 to h_6

for SEO

Starting tag must have end tag

each tag must be in lowercase

title must be used in head tag

heading tags h_1 to h_6 must be used under body tag

Google has chrome for search engine

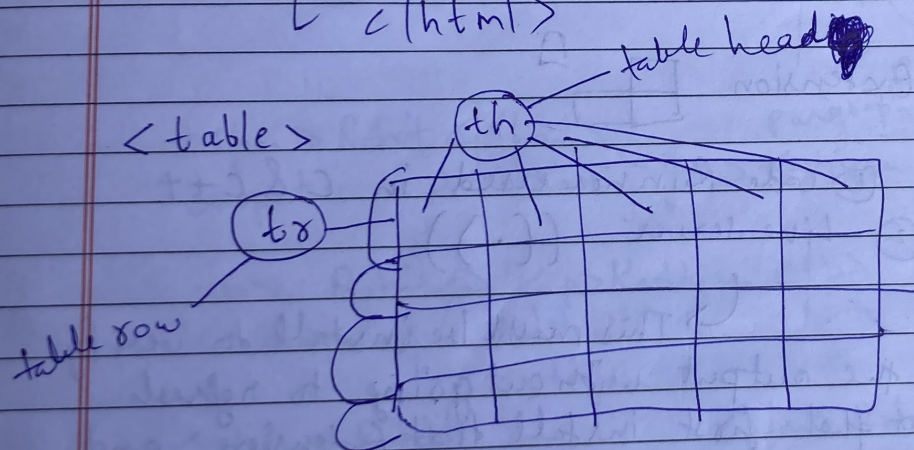
Meta data must be used under head tag

HTML file format

HTML file

```

<html>
  <head> <title> demo HTML </title>
  </head>
  <body> <h1> AB <h2> to <h3>
  </body> AB </h3>
</html>
  
```



</table> td → table data

<table>

```

<tr>
  <th> Name </th>
  <th> father Name </th>
</tr>
<tr>
  <td> Azadwindu </td>
  <td> Mr. </td>
</tr>
</table>
  
```


① Boiler Plate

complete HTML format

! → Ex net Observation

Visual Studio code

open desktop and create a new folder named anything to it and right click on it and select open with code that folder then click New file and write file name to anything with extension that HTML

① Extension

① Code Runner used in C & C++

② Live server ((.))

↳ This must be install to verify the output without going to refresh for that first install that Extension and click go live which is situated at the bottom right side corner then this will open automatically linked browser

for go live to be proper work internet connection must be needed

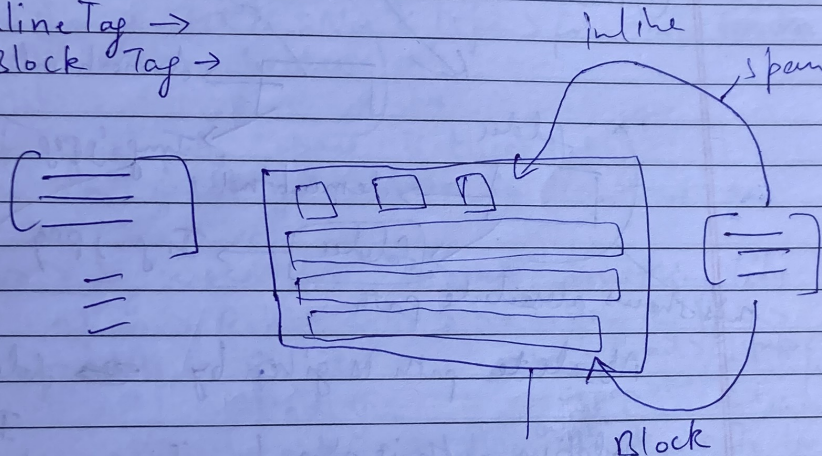
for Boiler plate to be work must install before live server ((.)) extension then open that file name with - Extension which you created then click shift + 1 you got it asks for Ex net Observation click that then you got complete HTML format called as Boiler plate

<!DOCTYPE html>

↳ this says about HTML5

(1) Inline Tag →

(2) Block Tag →



 Rohit Kadiyan
output is RohitKadiyan

<p> Rohit </p> <p> Kadiyan </p>
output is Rohit
Kadiyan

 Rohit

output is Rohit

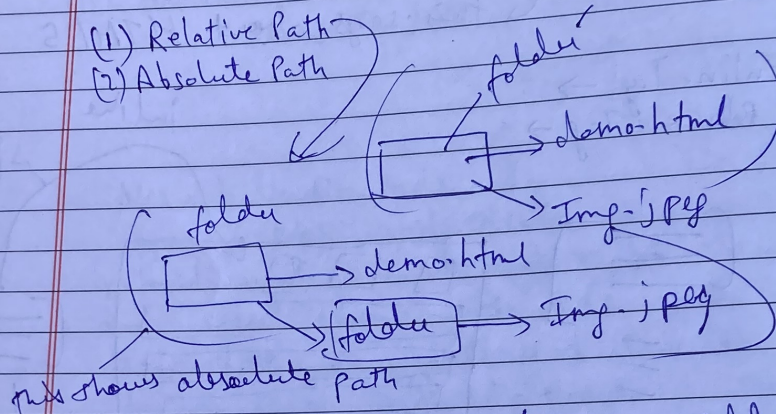
 Rohit >
 Kadiyan >
output is Rohit Kadiyan

Rohit
 Kadiyan
output is RohitKadiyan

Rohit
 Rohit

output is Rohit Rohit
Not's means give space
Not got RohitRohit

- (1) Relative Path
(2) Absolute Path



Absolute path is given by ~~folder~~ folder / folder /
 relative path is given by if you are in the
 folder where all files located then simply
 say img.jpg

(1) <HR> → use for horizontal line
 tag → has no ending tag

(2)
 → use for Break
 Tag → has no ending

mostly use in table as like bordering Tag
 and bottom like upper

for left and right side bordering use
 shift plus !, ! acts as |

` ^ SNo. ^ ` for left side and right side

`<HR>`

` <HR> SNo. ^ ` for Top ~~border~~ and bottom border

if you apply space between | |
then you'll get space in output too

`
` tag used for break something like

` Rohit
 Kadiyan `
output is Rohit
Kadiyan
` Rohit
 Kadiyan `
output is Rohit
Kadiyan

So, after break something text is to be written that comes in next line

→ has no ending tag
`<input type=" " >`

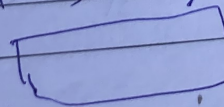
tag input tag has fixed attribute here is called type

input tag has no ending tag

How to apply attributes in HTML tag

`<input attribute1="value" / >` space attribute2
= "value" / space attribute3="value" >

attribute defines what is this tag
what work in tags

`<input type="text" >` value
output  a box you can write any text here

first attribute second attribute

`<input type = "text" placeholder = "what is your Name" />`

output is what is your name on typing only your text shows

placeholder is 'just write something inside box for just showing'

checkbox attribute for showing something like ☒ that's works on click on empty box as creating tick sign to that box

button attribute for showing something like this has click option so this acts as a button

`<input type = "button" />`

output is [scribble]

for using checkbox apply type = "checkbox" instead of text

~~name as the button~~

~~<input type = "button" />~~

if you want to show the left and right sides as at proper place like right side seems border like so we pre tag similarly this works for left side too

`<th><hr><pre> | SNo.`

output is

`| </pre><hr>
</th>`

other input tags are

`<input type="color">`

output is

filled color

default color is
black

when you click on filled color you can select any color

`<input type="date">`

output is

when you click on this box you can select date

`<input type="datetime-local">`

output is

when you click on this you can select date and you can see local time too

`<input type="email">`

output is

use this for writing email

`<input type="file">`

output is

No file chosen

use this to choose file

`<input type="hidden">`

output is ☐

you can click on that like button but when you give value by some text this works with button not with this

`<input type="image">`

output is

here we have image icon which to be valued not yet given any image so this is a icon

`<input type="month">`

output is

use this to select month

~~number~~

before typing

`<input type="number">`

output is



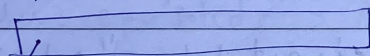
after typing



we have to write number

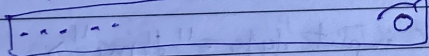
`<input type="password">`

output is

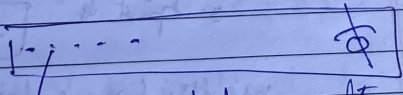


before typing

after typing



on click this



here is password shown after click

this



`<input type="radio">`

output is



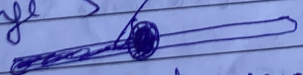
Black filled color after you

click this



`<input type="range">`

output is

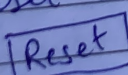


filled color

use this to increase or decrease some range

`<input type="reset">`

output is



use this to reset

`<input type="search">`

output is

before writing

after writing

`<input type="submit">`

output is

use this to submit anything

`<input type="time">`

output is

use this for select time

`<input type="week">`

output is

use this for select ~~the~~ ~~week~~ ~~of~~ ~~month~~
weeks as per the month

CSS → cascading style sheet

- ① Inline CSS
- ② Internal CSS
- ③ External CSS

`<starting tag style = "property: value;" href="#" >`
 such as border, color
 Green House is a property value
 Date Page No.
 attribute
 href="#" is a href attribute

href="#" is text
 to which whole CSS is
 applied

`<h1 style = "border: 2px solid red;" > A red </h1>`
 its width of border

Inline CSS means CSS is applied under any tag

`<h2 style = "color: green;" > This is the Heading </h2>`

Always style attribute is fixed to starting tag if you apply CSS, property and its value must be in lowercase

`<u style = "border: 2px solid red;" >underline tag </u>`
`<u style = "color: green;" >underline tag </u>`
`<i style = "border: 2px solid red;" >italic tag </i>`
`<i style = "color: 2px solid red;" >italic tag </i>`
`<strong style = "border: 2px solid red;" >bold tag `
`<strong style = "color: green;" >bold tag `

output is
 italic tag

underline tag
 bold tag
 black color
 red color

underline tag
 bold tag
 italic tag
 italic written

Other tags are ``, ~~``~~ ``,
`<video>`, `<audio>`, `<select>`

``

`AB`

`AB`

`AB`

``

for unordered list

output is

- AB that's
- AB mean
- AB output with dot sign prefix

``

`AB`

`AB`

`AB`

``

for ordered list, output is

1. AB
2. AB
3. AB

`<video>` → for ^{embed} video in HTML website
`<source src="ghi.mp4" type="video/mp4">`
`</video>`

`<audio>` → for embed audio in HTML website
`<source src="ghi.mp3" type="audio/mp3">`
`</audio>`

`<select>`

`<option>AB</option>`

`<option>AB</option>`

`</select>`

required

is used to select ^{required} item from number of items

option are used to what options are available during selecting

Output 4

ABV

When you click on arrow icon ∇ you get AB and AB as options for selecting. For selecting you have to click on the item which you want to select.

div tag acts as a container tag. It contains number of tags. If you apply any property on div tag then you get that same property on all tags which contained under div tag.
there are number of tags
 $\langle \text{div} \rangle$ $\langle \text{div} \rangle$

If you want to apply two properties with style then use single style keyword then write other property by given space.

$\langle \text{div style} = "border: 2px \text{red solid}; color: \text{red}; border-radius: 20px; width: 50\%; height: 300px;" \rangle$

Here are number of tags such as $\langle h \rangle$, $\langle input \rangle$.
 $\langle \text{div} \rangle$ all the property are contained under single quotes such as " $\text{width must be in percentage, height must be in pixel}$ ".
Do not apply semicolon at the end of attribute.

border-radius is another property to rounded the borders at corner. That's must be in pixel.

$\langle a \rangle$ tag anchor tag, if you write something in that's a file name.

That file has different number of page data so we linking that file name to other website page by given absolute path

for that we use keyword href
`` A file, which you want to
 path link end tag
 here is filename `` of anchor

if you are under folder admin, there's a filename called table.html

`Arad `

finally href is used to link one file to other, one webpage to other

if in case you give href a path # that's, means you can't exit current webpage even still you trying to move to other page by clicking filename or you click filename but you got same current page

if you use div tag then there number of tags be shown inline or their output break `
` tag after each tag at the end



`<div>`


`<input type="date">
`

`<input type="email">
`
`</div>`

Selecting a color from already developed website

right click anywhere at the chrome browser to the desired website then click on inspect left section for HTML and right section for CSS

for selecting a color you have to choose CSS section then find out this syntax, color: #1c1e21; like that, then click on filled color box then click on  filled color this icon then you get , move this to desired color place and click on that

in simultaneously, circle after  filled color

Now has desired color of desired place of desired website

that's comes under CSS section

Now, you got #1877f2 a desired color value of desired place

<button> tag

<button> </button>

Internal CSS ~~has three tags~~

~~① style tag~~

~~② id tag~~

~~③ class tag~~

syntax

```
<style>
</style>
```

Internal CSS has three types

- (1) Tag selector → It is used under tags if you apply an attribute, apply on all div's
- (2) ID selector → for selecting Id put id = " " ✓
- (3) class selector → for selecting class name put class = "anything class name here" ✓

ID be unique ~~for~~ so it can use once

But class also be used more than once
for Tag selector put ~~div~~ keyword to style tag
that's must be under head tag for applying any property to that

```
<style>
  div {
    color: red;
  }
</style>
```

for Id selector put first # sign then Id name to style tag.

```
<style>
  #head {
    color: red;
  }
</style>
```

Id name

for class selector put first . dot sign then classname

put under body tag

`</style>` class name

Green House
Date _____
Page No. _____

• head {
color: red
}

`</style>`

`<head>`
`<style>`

div { background-color:
black;

win {
color: blue;

h1 {
color: blue;

• winner {

`</style>`

`</head>`
`<body>`

`<div>`
`<h1>Azad</h1>`

`<h1>Azad</h1>`

`<h2 id="win">Azad</h2>`

`<h3 class="winner">Azad</h3>`
`</div>`
`</body>`

background properties

`<body>` class selector
`<div class="Azad">Azad</div>`
`</body>`

`<style>`
div { width: 100px;
height: 300px;



ate - India's No. 1*
is now a Superbrand.
MRB in Jan 2019.

border: 8px solid red;
background-color: red;

this may be relative
or absolute

• A red ✓
background-image: url("file name");
background-repeat: no-repeat;

we use div selector because we want to use container and class here to use background image automatically images ~~automatically~~ in internal css div width and height as per the selector applied to it.
background-repeat: no-repeat; that's means there is no repetition of one pic more than once on single webpage

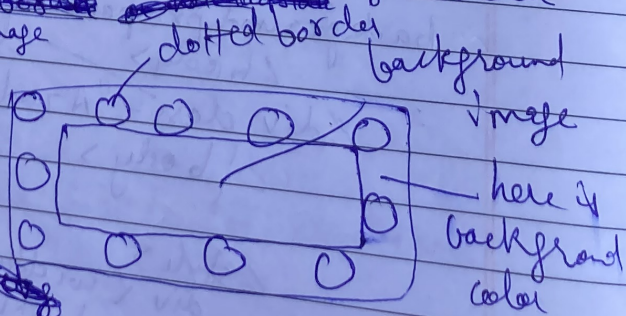
background-size: 100% 100%;
we use this to present background image width and height

here first 100% is for background image width and other is for height

background-clip: border-box;

we use this to put background image under border not on border ~~on border~~
div before that we put change for image

output is



~~we use this to border~~

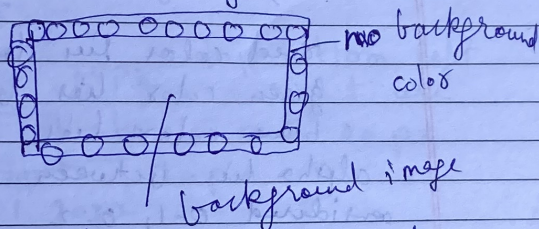
so that you can see background color

background-clip: ~~border-box~~ content-box;

we use this to put border outside div

we use this to hide background color

output is



generally in dotted border we get background color ~~that's why~~

background-position is used for background image position where to be located either at left right top and bottom

background-position: center;

background-position: right;

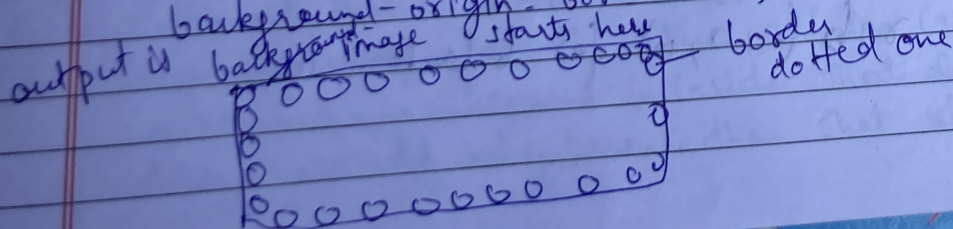
background-position: left;

background-position: bottom;

background-origin: content-box;

is same as background-clip: border-box;

background-origin: border-box;



rgba stands for red green blue alpha

rgba (red percentage here, green percentage here, blue percentage here, alpha)

also, a rgb available

red color lies between 0-255

green color lies between 0-255

blue color lies between 0-255

alpha lies between 0 to 1 (here also considered 0.1, 0.2)

alpha keyword shows opacity or transparency of the color that's means more the opacity we can watch through that what is behind

if we try to lesser the transparency color goes solid

rgb(255, 0, 0) for red color

rgb(0, 255, 0) for green color

rgb(0, 0, 255) for blue color

If we try lesser the value from 255 to lower the color getting fades away on lowering the value

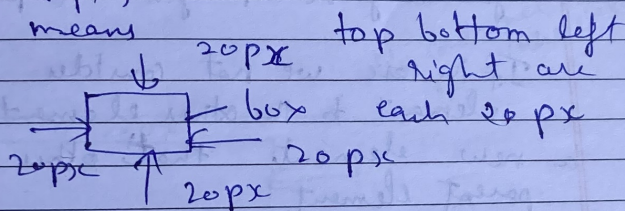
rgb(0, 0, 0) for black color

Margin is the space from the browser boundary to the div outside

padding is the space from div to inside (that may be box or other)

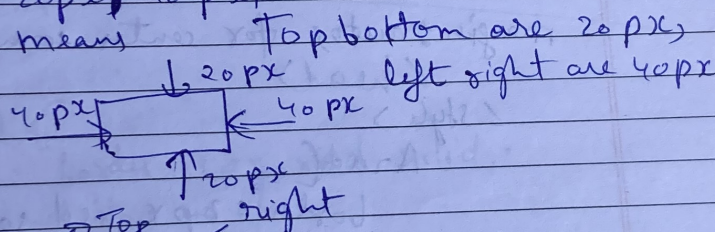
① Margin: 20px;

that's means



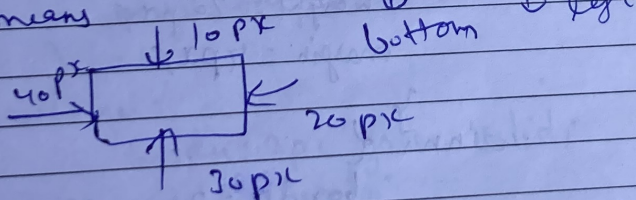
② Margin: 20px, 40px;

that's means



③ Margin: 10px, 20px, 30px, 40px;

That's means



Browser is actually is a parent element to the ~~other~~ other which is inside, so for browser that element which inside ~~browser~~ parent element that's cover 100% value and browser covers 100% value, if in case a other new element in inside that element which is already under parent then that element

is also 400% for approaching the parent element ^{not}

parent element ^{be 100%} for responsive website ^{we should give small element to the 100%} and we don't touch that element width and height, if you want to do that you may use margin and padding

If in case we not consider browser as parent element to the other element then for a new element that other element acts as parent element

margin: auto; for center something

<head>

<style>

.Aradf

border: 2px red solid;

width: 400px;

height: 20px;

margin: 4px;

- winner &

border: 2px green solid;

width: 100px;

height: 4px;

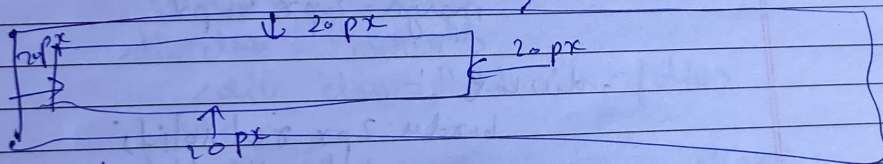
</style>

</head>


```
<body>
  <div class="winner">
    <div class="Arad"></div>
  </div>
```

we here takes winner class as parent class to the Arad class

output is



```
<head>
<style>
```

```
.Arad {
  border: 2px red solid;
  width: 400px;
  height: 20px;
  margin: 4px;
```

```
.winner {
  border: 2px green solid;
  width: 400px;
  height: 50px;
```

```
-div1 {
  border: 2px green solid;
  width: 400px;
  height: 50px;
```

```
}
```


• div2 {

border: 2px green solid;

width: 40%;

height: 50px;

}

• div3 {

border: 2px red solid;

width: 40%;

height: 20px;

margin: 20px 40px;

• div4 {

border: 2px red solid;

width: 40%;

height: 20px;

margin: 10px 20px 30px 40px;

}

• div5 {

border: 2px green solid;

width: 40%;

height: 50px;

}

• div6 {

border: 2px red solid;

width: 40%;

height: 20px;

margin: auto;

}

</style>

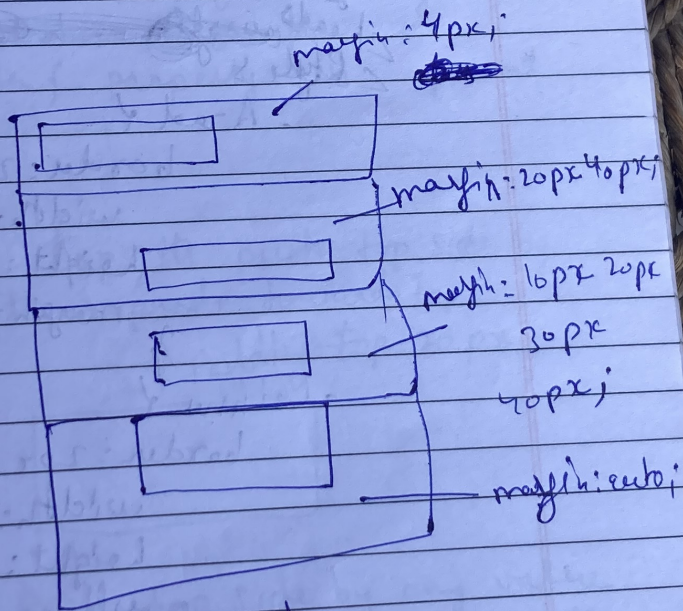
</head>


```

<body>
  <div class="winner">
    <div class="Azed"></div>
  </div>
  <div class="div1">
    <div class="div3"></div>
  </div>
  <div class="div2">
    <div class="div4"></div>
  </div>
  <div class="div5">
    <div class="div6"></div>
  </div>
</body>

```

output is



margin-top: 20px; inside
push element 'div' from top towards
bottom by 20px

margin-bottom: 20px;
push element inside div from
bottom towards top by 20px;

margin-right: 20px;
push element inside div from right
towards left by 20px;

margin-left: 20px;
push element inside div from left
towards right by 20px;

overflow: hidden;

~~head~~
~~<style>~~

Arad

border: 2px red solid;
width: 1000px;
height: 200px;
margin: 4px;

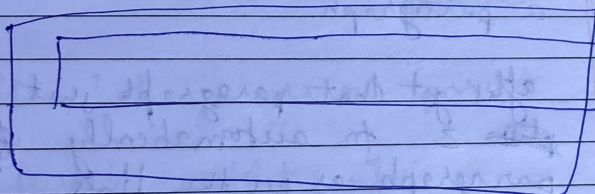
win

border: 2px green solid;
width: 400px;
height: 50px;
overflow: hidden;

</style>
</head>


```
<body>
  <div class = "winner">
    <div class = "Azed"> </div>
  </div>
</body>
```

output is



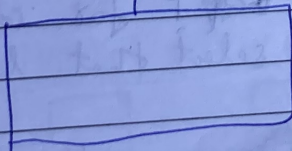
that's means class Azed, hided by class winner so we use overflow hidden property in parent element

Padding

padding-top; its push top side by any value or its expands towards top

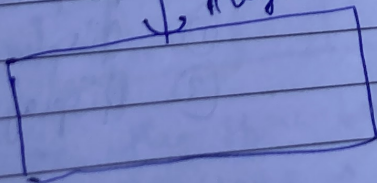
↑ padding-top: 20px;

output is



margin-top; its push top side by any value towards bottom but not impact on other sides

↓ margin-top: 20px;



font-size: 12px;
which is used to increase or decrease the font size, this is applied to the div which contains that font

lorem is a inbuilt html and css paragraph for use. type lorem and then enter and you get a paragraph

after get that paragraph just ~~ctrl~~ type ~~ctrl~~ ^{alt} z to automatically adjust all paragraph as per the lines

Control z for backspace

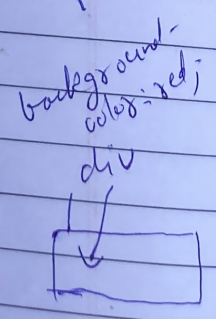
Control s for save all the things what you write

Alt tab for easily switch from current tab to website or website to tab

Control ~~alt~~ P for comment
for use select that line use this control P

Display ~~flex~~ properties:

- ① Display: none;
- ↓ Properties value
- ② Display: show;
- ③ Display: flex;



① Display: none;
next hides only background but
also the div

② Display: show;
if you applied already display
: none; display: show that
content that again shows

③ Display: flex; always use on parent

① Justify - content: center;

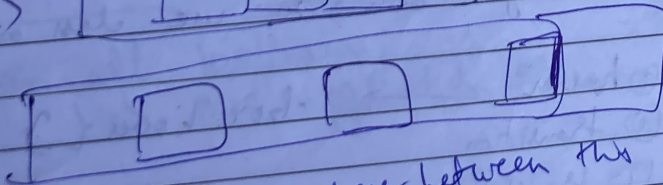
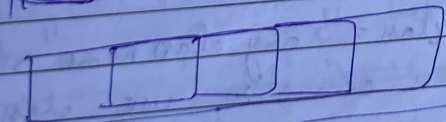
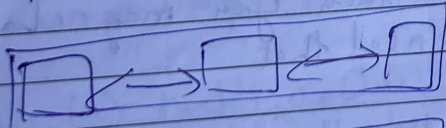
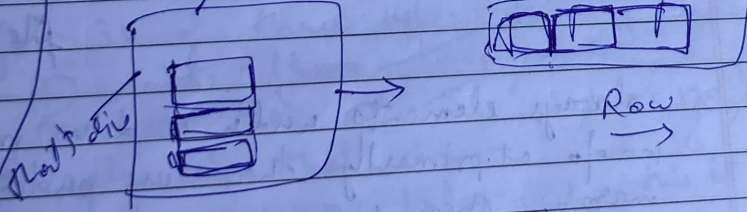
② Justify - content: space-between;

③ Justify - content: space-around;

These three properties can't work without
display: flex

Display: flex; convert column to
parent column

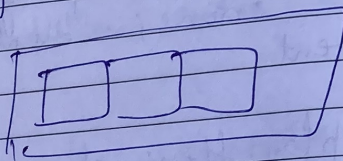
row



if actual is 10px after space-between this do be 100px
so we have to take 100px as 9px


```
<div> box1  
    box2  
    box3 </div>
```

output



`text-align: center;` to put the text at the center

`border: none;` to remove the border

``

↓
file path → `href`
C/A

always elements under a div must be of 100% at primarily then use padding or margin then you may change width of that if need that

`<body class="any class name">`

you can also use class on body tag




① hover

② transition

③ transform

→ `.box: hover { }`

Three properties of cursor

cursor: pointer; output is 
 cursor: zoom-in; output is 
 cursor: zoom-out; output is 

transition can't work without hover

transition is to control the speed of the .box
 hover

we use linear keyword to say anything going
 with same speed

transform is to rotate the box

↓
 property ↓
 its value

Transform is of three types ^{such as 120, 180, 90} degree enter here

① rotate ~~for~~ degree enter here

② ~~skew~~ skew (^{such as 120} degree enter here)

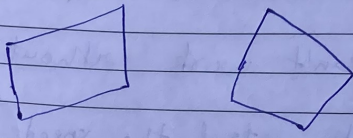
③ scale (1) that's same as 0px 4px before and after ^{this is laptop type}

speed in seconds

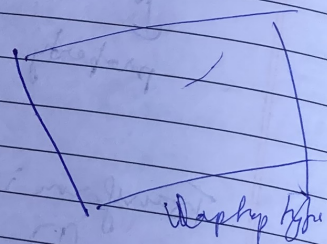
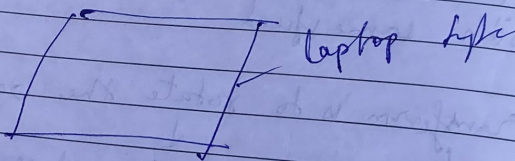
scale (2) that's doubled the values
 such as 0px ^{width} 8px ^{height}

transition: all 10s linear;
 ↓ that's applied in ~~the~~ same proportion on whole

rotate transform outputs are



skew transform outputs are



transform: scale (2);
that's doubled the values

we have to use transform under hover
such as .box: hover {

transform: scale you can do on .box { }
hover it to apply properties

transform: rotate (120deg);
transform: rotate (90deg);

→ this used for if
~~the~~ horizontal line
to you have to
rotate, for this, use transform

shows like this
(1)

rotate is to slight rotate the things
~~space is to rotate~~

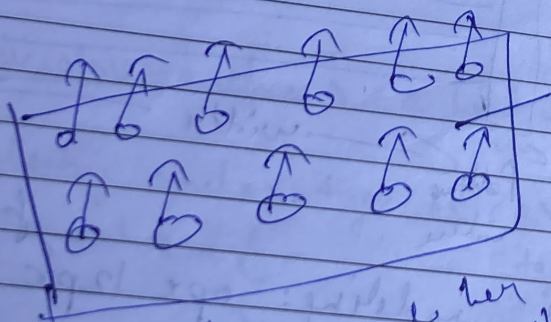
for using box: hover

we first create box { with border width & height

for hovering give background color to box: hover

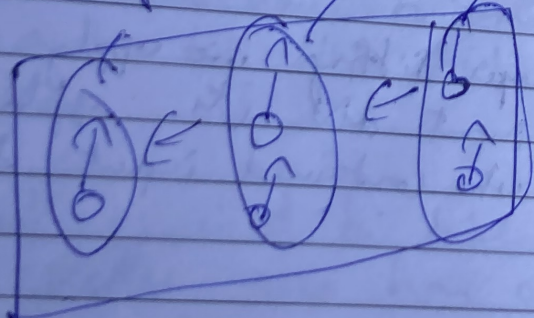
At primarily don't use padding
use under div must be of 100% width
not to reduce that

linear ~~and~~



color proportion changes as same in wheels

case word her second her first her

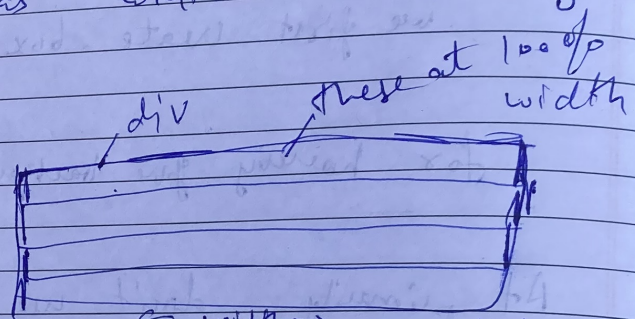


color proportion changes in steps

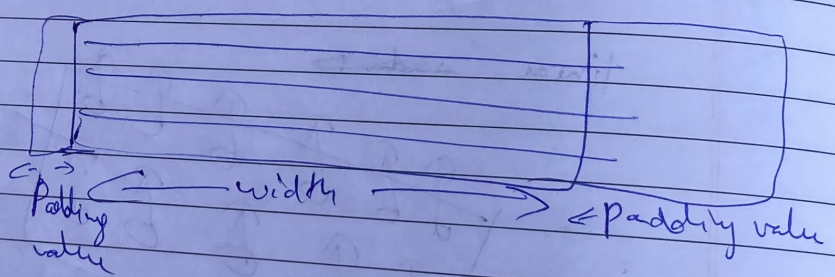
padding is background box

padding is width of the box ~~div~~ to whom you applied padding and padding value itself

such as width value + padding value
output



on applying padding background box created



padding: 0px 12px for left and right

for proper fitting we apply padding not on input div its applied on outer div who contained input div

for responsive website and we use padding after width of inputs to be too do then we do work on inputs div then apply padding on outer div which contained inputs div

class selector which is something absolute path to specific class

such as

```
<div class="box1" >
  <a href="#" > </a>
</div>
```

not to class of that tag

for selecting anchor tag or <a> tag
use .box1 a {

```
<div class="anchor" > <a href="#" > tag in
  </a>
</div>
```

for selecting anchor tag or <a> tag
use .anchor a {

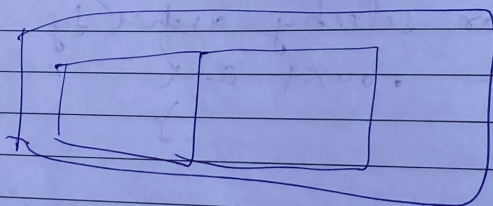
text-decoration: none; to remove underline from anchor tag link such as forgotten password

screenshot tool is used to take screenshot of the selected area from any website

mostly uses for logo
for use this search shipping tool in search
bar then click windows icon + shift
+ S then draw or outlined a
specific logo or image to whom you want
to save after selected then click ...
three dotted line and click save

for easy movement of the input tag use div or
that

padding works same as margin

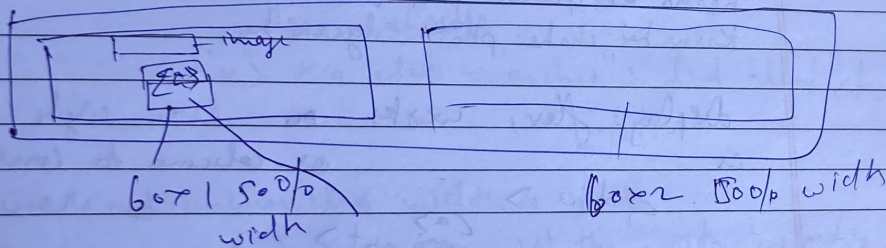


we want to shift left box to right
side and right box to right side.
so we have option of margin-left for
left box and margin-right for right box
in opposite we may use padding
can do both works such as ~~padding~~ margin-left
and margin-right for both the boxes

How is to use cursor: pointer property on the website or its related website ka

Sabse pehle design or structure tayyar Kiya jata hai fir write Kiya jata hai text ko

such as 100% width div or box



We try to use div less so we may use three or more anchor tag under a single div but note that we do use color properly on the anchor tag so that we may use different div for each anchor tag

after create structure then we write text or elements because if we write text before structure created that's increase the height of the box where written

Kindly note that we use div as a this tag to be come under display: flex; property for converting column to row

We have to create whole structure whole structure but partial one

If you want to give margin from left
or to right you may use margin: auto to
as (or say both)
center this

सबसे पहले नीचे वाली है दिव लगाने
फिर नीचे बहरी है बोर्डर लगाने फिर इन
नीचे के गुदा जो भी काम करना है वह
करना है चाहे photo लगाने है

display: flex; works on even style chain
as or column to convert

<div> to row
<input>
</div>

so we

if you apply width: 300% as that's
width already given to the button tag

with the button tag on short the
page the button icon goes out from
own layout so we apply that width to
the div which applied on that button
tag and in that case button must be of
100% width

if you take div <div>
then you have to take another div
on that now inner div must be of 100%
~~on that~~

and enter div of ~~as~~ you need
even on anchor tag text as

```
<a class="anchor">find talent</a>
```

so you have to div on that
take another
class= anchor512

```
<div><a class="anchor">find talent</a>  
</div>
```

now anchor class width is 100%

and you now set the width accordingly
to anchor512 as width: 4px;

display: flex; not applied on another div its
applied on real div as

```
class="anotherdiv"  
<div><div class="real"></div></div>
```

you applied on real class div as you need

padding

if

Search

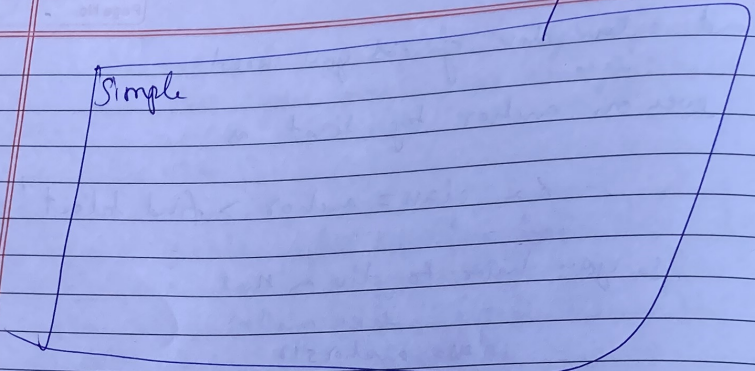
after 4px left padding output is

Search

texting start from here

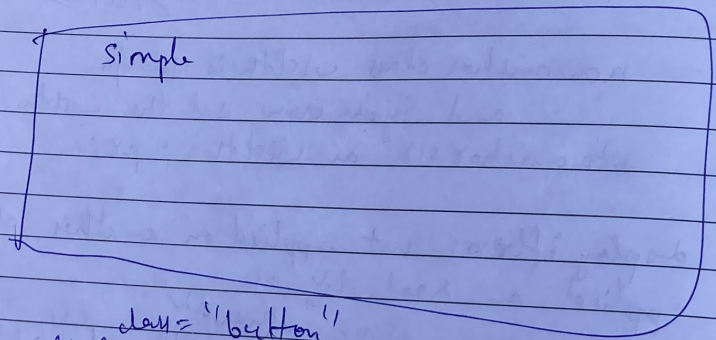
/607

Simple



after padding - left: 4px

Simple



class = "button"
<div> <button class = "button"> Sign Up </button>

you have to apply background color on button not on another div </div>

height is applied on real div in all the cases not on another div

<div class = "anotherdiv"> ^ <div class = "button">
you have to apply height on button class div not on another div class

in the case of anchor tag $\langle \text{div class} = \text{"anchor"} \rangle$
 $\langle \text{div class} = \text{"another div"} \rangle \langle \text{div} \rangle$

~~you have to supply first size on~~
~~another padding not on~~
~~another div~~

With padding-left: 8px; width increases
 so you have to sure reduce width
 according to need too

~~with padding-right: 8px; to~~

With padding-top: 8px; ~~width~~ height increases
 so you have to sure reduce ~~width~~ height
 to need too

after padding-top: 8px;

Search

before

Search

Ctrl plus F for search any keyword
 across the project

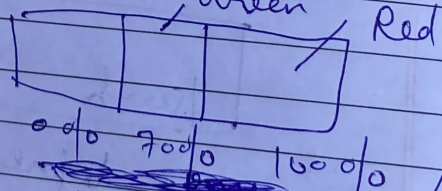
if you use ^{pre under} paragraph tag then you have

to select pre tag for further
subas

```
<div id="pre">
  <p> <pre> <code> </pre> </p>
</div>
for this
<style>
  #div1 pre {
  }
</style>
```

Animation is a property for adding this to
your css, you have to add ~~the~~ `<style>`
`</style>` tag

@keyframes my ^{anything} name ^{here}
0% { }
50% { }
100% { }



~~animation~~ ~~name~~ ~~value~~

@keyframes my ^{property} ^{its value}
0% { Background color: Blue; }
50% { Background color: green; }


```
100% { background-color: red; }  
}
```

```
<style>  
  .box {  
    animation: my all linear 1 2;  
  }  
</style>
```

Same speed of whole once
animation

~~1 2~~ → twice animation

```
<div class="box">  
  content  
</div>
```

0% → 100%

for apply animation

first add this

```
<style>  
  .box {  
    animation: my all linear 1;  
  }
```

anything name here once animation

```
@keyframes {  
  0% { background-color: blue; }  
  50% { background-color: green; }  
  100% { background-color: red; }  
}
```

anything name here

animation not applied on image its
applied on button or on .box

<style>

button animation: my 2s linear 4; after
repeat;

@keyframes my 1

</style>

0% border: 1px solid red;
background-color: blue;

Delay time
I mean animation
starts after how many
seconds on
repeat;

① Google font

Search google font on google search engine
and you have got website similar to this
<https://fonts.google.com> then choose whether
you need to use fonts or icons, so here
we are selecting and searching fonts so click
on that and now click font families such as
Roboto, Poppins, even you can search font
family in search bar such as (Search fonts) this
Now scroll down and select font type whether
its thin or italic by clicking this select thin
100 (+) and this get added to your this
option ☐ + now click this ☐ + and get a
code to embed this to your ~~html~~ html
file as under head tag ~~such as~~ such as
<link ~~href~~ href="https://fonts.googleapis.com/css2?family=Poppins" style="font-family: Poppins;"
>

under
styles
tab

then add this to your css file under style tag

```
.box {
  font-family: 'Poppins', sans-serif;
}
```

```
<body>
  <h1 class="box">Arad</h1>
</body>
```

② font awesome ^{use} for use icon before and after text that's be under body under tags or before and after ^{hyperlink} hyperlink

```
<body>
  <h1 class="box"> fontawesome, space (Arad) </h1>
</body>
```

such as google icon ^{of type which you apply} ~~that's not~~
 <i class="fa-brands fa-google"></i>

other thing is use cdn of font-awesome for working of whole font-awesome that's used under ~~font~~ head tag such as ^{for} link ^(or abbreviation is </> for copy and paste to your file) for cdn of font-awesome, search cdn of font-awesome and you get cdnjs.com/libraries/font-awesome and do copy and paste

③ External CSS

Inline CSS	Internal CSS
style tag	style tag
as	as tag used
at the bottom	under head
used	tag

External CSS
has no style tag
we have ~~create~~ ^{create}
new file with .css
extension for adding
css rather than under style
under head tag

CDN is content delivery network
http is hypertext transfer protocol

External CSS

file(1). external.html
file(2). external - CSS.

in external CSS has no head of style tag under head tag

after that we link the - CSS file to -html file
so we used this

add this under your head tag

< link rel = "stylesheet" href = "anything" style = "border: 2px solid red; border-radius: 40px;"

and .css file

that's standard

file name here with - CSS extension

and under body box

<div> </div>

And

if you take percentage under @keyframes you should be take as under sequence such as

@keyframes my {

0% {

100% {

200% {

300% {

up to

1000% {

}

for using images under animation use

@keyframes my {

0% { background-image: url("8pd.png");

background-size: 400% 100%;

<body>

<div class="box"> </div>

for width
for height

</body> <style> </style> <head>

. box {

animation: my 2s linear 1;

border: 2px solid red;

height: 200px;

width: 300px;

}

that's necessary to use after
background-image ~~background~~

in keyframes under percentages
such as under 0%.

@keyframes my {

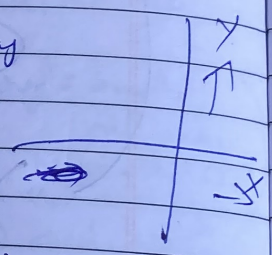
0% { background-image: url("8pd.png");

background-size: 100% 100%;

}

Bootstrap 4 → 5

- ① cdn
- ② start template
- ③ $l\text{-sm}-1 \rightarrow 8.3\%$ width $\langle \text{fluid} \rangle \langle \text{div} \rangle$
 $2 \rightarrow 16.6\%$ width
 $\text{class} = "col-sm-1"$
- ④ border
- ⑤ container (90%) IInd container-fluid (100%)
- ⑥ $m\text{-x}-\text{auto}$
- ⑦ $m\text{-t}-12$ $m\text{-b}-12$ $m\text{-s}-12$ $m\text{-e}-12$
- ⑧ colors \rightarrow text-succes, primary, danger, light and dark
- ⑨ Background color \rightarrow bg-succes, primary
- ⑩ $m-7$ apply on all sides



bootstrap is framework of CSS

~~we~~ already made class that's been used by some shortcuts or keyword to direct access that class not to create that one

currently bootstrap 5 is going on
for using bootstrap in your project we do use starter template rather than boiler plate because starter ~~plate~~ template is responsive

for obtaining ~~bootstrap~~ starter plate search introduction of bootstrap and you got page like <https://getbootstrap.com/docs/5.0/getting-started/introduction/>, now scroll down and find starter template then copy and paste in your project, as we know ~~a~~ link ~~a~~ tag is already

included in this ~~bootstrap~~ starter plate, so we need to add again link tag in the head tag don't

actually bootstrap works in html file extension so we to create anything file name with .html extension, now remove unnecessary files which you don't need such as script src

bootstrap also uses as class under body tag but don't need to create something like ~~box~~ ~~border~~ CSS

for here we use just keywords like border, bg for background, we to use these keywords under class under body tag

for multiple properties class to look like this

```
<div class="border m-4 mt-12">  
  </div>
```

but something is that we to use class under div

for searching CSS terms in your bootstrap website use Introduction of Bootstrap

webpage or getbootstrap.com/docs/5.0/getting-started/introduction then search for

search class bar and enter CSS property which you want to use

col is for column
col-sm-1 is standard with width 8.3 o/o
2 width 16.6 o/o

border is used for border property
container is used for div with 90 o/o width
container-fluid is used for div with 100 o/o width
mx-auto is used for margin auto across
horizontal or x-axis

my-auto is used for margin auto across
vertical or y-axis

mt-12 used for margin-top: 12px;

mb-12 used for margin-bottom: 12px;

mt-12 used for margin-top: 12px;

ms-12 used for margin-start / margin-left: 12px;
margin-right: 12px;

color → text-success for ~~blue~~ green color text
text-primary for blue color text
text-danger for red color text
text-light for white color text
text-dark for black color text

bg-success for green color background with
bg-primary for blue color background with
bg-danger for red color background with

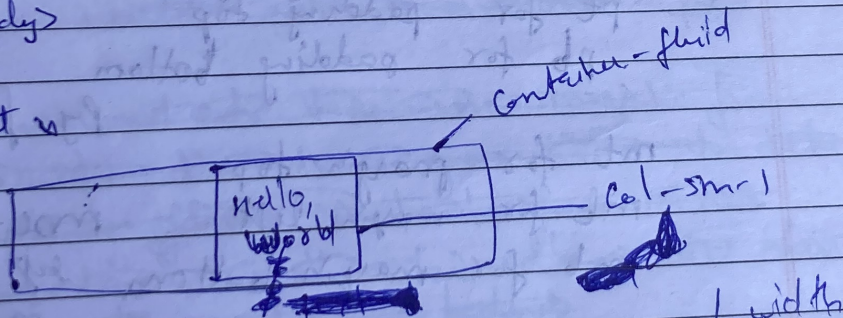
m-4 used for margin: 4px; applied from
all the sides
p-4 used for padding: 4px; applied from all
height property is manually change there's no
key word related to that in bootstrap

pt-12 used for padding-top: 12px;
pb-12 used for padding-bottom: 12px;
pl-12 used for padding-left: 12px;
ps-12 used for padding-start / padding-left: 12px;

mt-5 has final limit is 5
so its can't work on mt-6

```
<body>
  <div class="container-fluid border bg-primary">
    <div class="col-sm-1 border bg-danger mpx-
      auto"> <h1 class="text-danger bg-light mt-2">
        Hello, world! </h1> </div> </div>
  </div>
```

output is



for demanding width

With bootstrap if you take <div class="col-sm-1"> </div> and put that
so you have to take div on that <div class="col-sm-1"> </div> and put that
<div> <div class="col-sm-1"> </div> and put that
class="container" </div>

~~If you want to take height 100% or so~~

~~set the height to 100~~
~~height: 50px~~

~~generally height with 50px or width with 50px and similarly to top and bottom~~
~~width and height~~

~~for input width to 100px~~ In bootstrap there is
~~add width~~ no width and height property

col-sm-1 goes upto 14 such as col-sm-14
 margin of all sides top, bottom, left and right goes upto 5 level, similarly level 5 to padding

pl for padding left
 pt for padding top
 pb for padding bottom

px - for both sides left and right
 py - for both sides

mt for margin top
 ml for margin left
 mb for margin bottom

top and bottom
 mx - for both sides left and right
 my - for both sides top and bottom

font-weight - bold for bold the font-type
 btn-primary for button color blue

text-center for locating the button at center and text such as headings at center
 d-flex for display: flex;

btn - success for button color green

~~display-1~~ works as ~~h₁~~ font size
~~display-2~~ works as ~~h₂~~ font size
~~display-3~~ works as ~~h₃~~ font size
~~display-4~~ works as ~~h₄~~ font size
~~display-5~~ works as ~~h₅~~ font size

if want to increase col-sm-1 then you have to take div on that to increase that col-sm-1 upto 14 such as col-sm-1 without that that's not work

~~height in percentage not generally works so you have to take height in pixels even if you want to take 100% of the page~~

~~if you want to take no margin from body to div then you have to take container fluid instead of container~~
~~is <div class="container"> </div>~~
~~col-sm-1 </div> </div>~~

if you have anchor tags so you have to div take like this

col-11 class="container">
 <div class="col-sm-1"> </div> </div>

for 1000% width use container-fluid
in div class
for 90% width use container m ddb
class

for demanding width use col-sm-1 upto
14

col-sm-1 width 8.3%

2 16.6%

3 24.9%

4 33.2%

5 41.5%

6 49.8%

7 58.1%

8 66.4%

9 74.7%

10 83%

11 91.3%

12 99.6%

13 107.9%

14 116.2%

0

if in case you are in the third div
<div >

<div >

<div class="div3">

</div>

next's

you are working under
first div so don't touch

</div>

</div>

other div's

in bootstrap we prioritize to use less
div even still the webpage is automatically
responsive

previously was different

But we do take div & then do
other div to that such as

~~other thing be changeable and seal
div has width of 100 to be sure~~

But now that's not need to
such as on all properties majority
such as button input heading paragraph
also we can't have too div's on that
take

real
> (div) (div) </div> </div>

for why display: flex;
must be under div's
each tags

such as <hr> tag, <p> <input>
tag and may more
such as <div> <hr> </div>
<div> <p> <p> </div>
<div> <input> </input> </div>
</div>

max-auto for center the box not applied to text

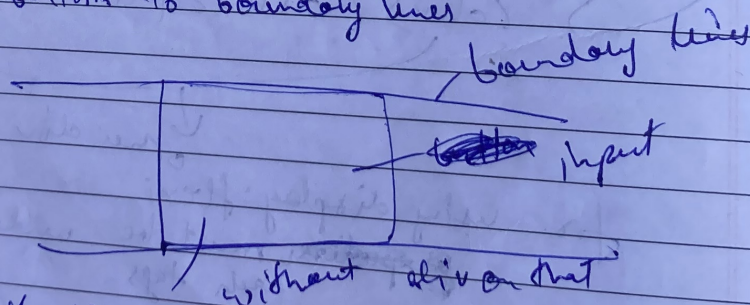
for text's use text-align

in bootstrap you can use margin sides even you applied margin-auto, here it is

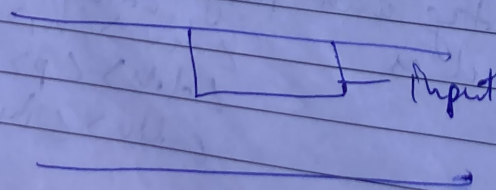
max-auto
such as $\left(\begin{matrix} \text{max-auto} \\ \text{mt-5} \end{matrix} \right)$

but previously in CSS that was not working

sometimes we takes div in a optional way its reduces the height of the components to 50% otherwise components touches both top and bottom to boundary lines



after div that



bootstrap

Green House
Date _____
Page No. _____

In manual scale is important to structure the website

such as take a 100% line either horizontally or vertically

→ 100%

now put the relevant components on that

90% 20% 40%
| |
→ 100%
div

now setting the components percentage on that
and you may adjust the percentages according to the need

40% 18% 41%
| |
→ 100%
div

same goes with vertically

↓
100%
↓

100
82
18

41
41
82

initially a div automatically takes some padding, you have to put padding $p=0$ for no space ^{so} from outer container to inner container

In bootstrap we to use less div and classes

bg-secondary used for light black color

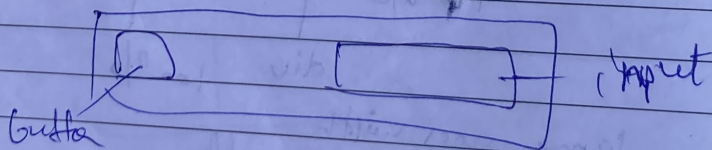
bg-warning used for yellow color

bg-info used for light blue color

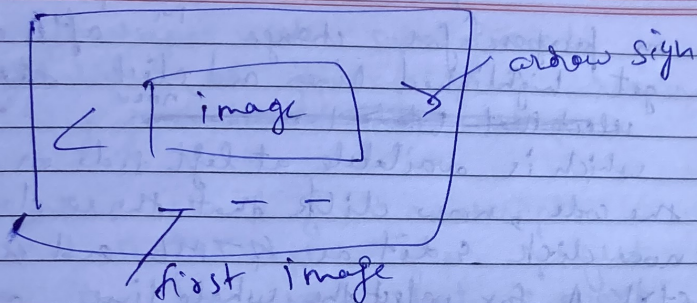
bg-white used for white color

(1) Navbar commonly known as header but it's of its type

Navbar has button input ~~as~~ as per need you can add




(2) ~~Carousel~~ Carousel its like slide show number of images but it has option for you can shift from one pic to other by arrow signs < >



(3) Snippet is already made code for specific purpose such as login form but can't use them directly for the reason of you duplicacy so you have to apply changes to that such as change button text and color and background color as per your need

for searching snippet of code just type snippet of login form in bootstaps means login form here purpose In either case bootstaps code can't run into bootstaps, so you have to search according to that

after get code copy that and paste your project if your project of html that's be pasted under <body> tag now run that

as you want to change text of button as sign in just right click that and click on inspect now click on this icon  on click that's must be turn blue after that click

such as how sign button
on purpose for change in after click you
get highlighted line and click ~~ctrl A~~ for
~~select that whole line~~ now ~~...~~ (three dots)
which is available at left side or start of
the code, now click that three dots and
now click edit as HTML and now click
ctrl A for select the whole line and ctrl C
for copy that, now move to your project
now click ctrl F for search something

This way fast your work rather than searching
whole project

In meanwhile button can't work without
javascript that is a mechanism which handles
everything behind the bootstrap

Bootstrap 3

Bootstrap 4

Bootstrap 5

Container size 1170px

1140px

1320px

column's available in sizes
are sm, md, lg, xl

column's available in
sizes are sm, md, col,
lg, xl

column's available
in sizes are sm,
md, lg, xl, xxl

Non-responsive layout

Responsive
layout

Responsive
layout

Bootstrap is the most
stable version of
bootstrap

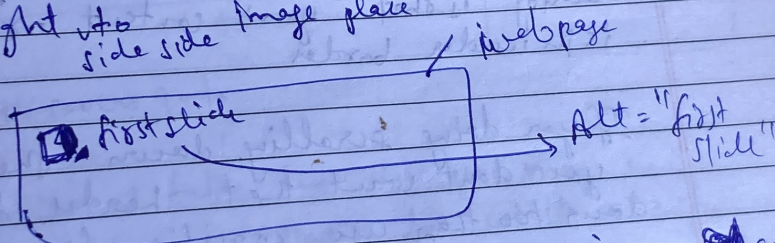
Bootstrap 4 is a
newer version of
Bootstrap with new
components and with responsive
layout

Bootstrap 5 has
smooth interface
here jquery
replaced by a
javascript

<script> tag with src of cdn or code of jquery or jquery is the reason of working of buttons in bootstrap

js here stand for javascript
if that that script in your project the button may be seen but not working at all

Alt attribute under image that's a specifying text showing image of which type if your internet is slow or not working image may be not loaded in that case you get that written text on your web page just right into side side image place



In bootstrap 4 script uses are code: jquery, cdn. jquery (maybe 2 or more)

In bootstrap 5 cdn of jquery is necessary to use (maybe 3 or more)

That's be used under body tag
in prebuilt starter template has already these scripts to bootstrap version such as bootstrap 4 or bootstrap 5

if you applied margin-bottom this may increase height so you have to use height manually by CSS as per need such as

<head>

height { height: 200px; }

</head>

In bootstrap if first applied padding such as p-0, if need then remove the border such as border: 0; then now apply margins such as border: 1px solid #000; margin-top, bottom, left, right; then again apply border: 1px solid #000; border-primary for clarity is that your content or components is in under border

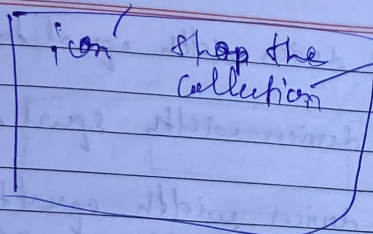
If you doing scrolling down the page and you don't want that header to be scrolling down too then use position: fixed; properly

if you use font-awesome icons that's not working or sometimes not visible then convert regular to solid such as in <i class="fa-light"> </i> to <i class="fa-solid"> </i> same for regular one

border goes from 1 to 5

Another property if you have situation like

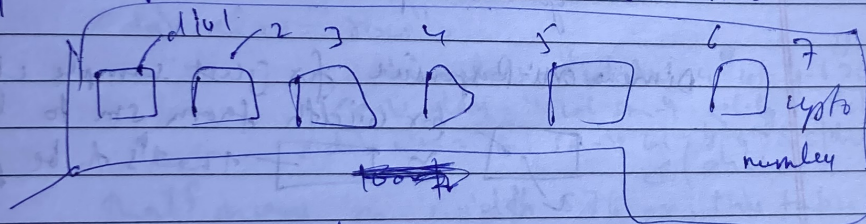
Header



It should be
start behind the
shop then
use span tag
on collection and
give that ~~class~~
name's margin

`<p class = " " > Shop The
`
`Collection </p>`

for responsive website these are clarity's

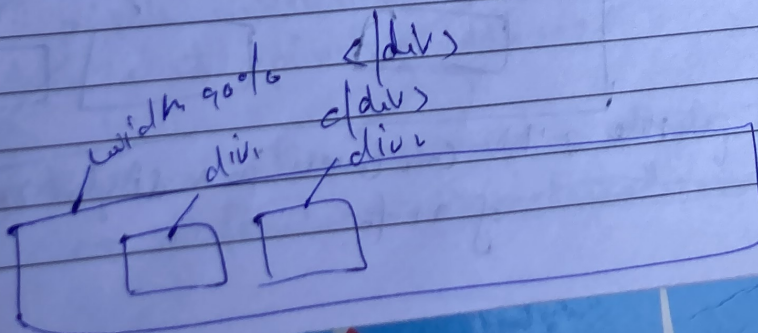


this width covers 90%

if you want take less div you may use
col-sm-3 such as, col-sm-4 for width-90%

if you want to take number of columns as in
single row you may use `class = "row" > <div>`

such as `<div class = "row">`
`<div class = "border col-sm-3`
`col-lg-6">`






col-sm for devices width equal to 768 px

col-sm for devices width equal to 576 px

col-lg for devices width equal to 992 px

col-xl for devices width equal to 1200 px ^{as same}

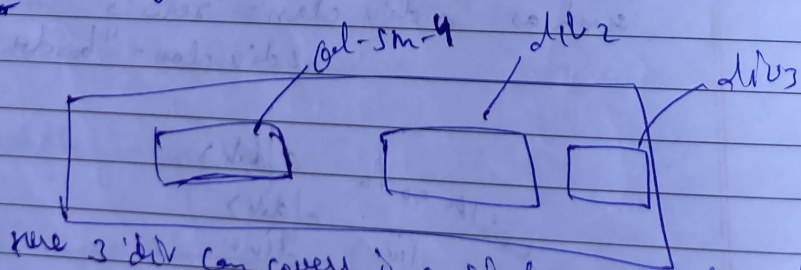
~~for responsive website~~ we use some tools such as ^{for font not come, get from fire button} go on inspect and click this I can 

Dimensions: Responsive for just sample checking for width from sm to lg, xl
 x  that's to be put same

change this according to need like on 992 px width, col-lg applied to webpage

here, col-lg-6 means large columns but that's also take width of the small one such as col-sm

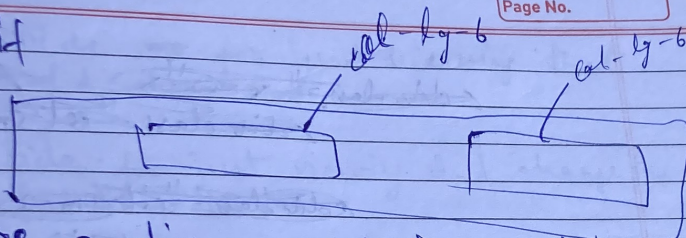
col-sm-6 for 49.50% width that's also applied on col-lg-6 ~~but that's not that's means~~



here 3 div can covers in a single row of width of 90%

previous font-size we don't change but or may more

if



Here 2 div can covers in a single row of width of 90%.

if you applied both on same div that means div has both properties such as

`<div class="row">`

`<div class="col-sm-4 col-lg-6">`

`<div class="col-sm-4 col-lg-6">`

`<div class="col-sm-4 col-lg-6">`

That's means on width 576 px, this takes 2 div across a row.

but simultaneously if you increase width to 992 px by ~~adding the width~~ ^{single} 2 div's can cover by a single row and other div is a that's a responsive, basically kept row for increases or decreases randomly height of one.

That's means at large column stay at width 99.5% such as `col-lg-6` for 992 px width.

You have to design start from `col-sm` at inspect pixels width 576 then set the `col-lg` which is at pixels width 992.

~~you have an alternative way to display this~~
~~or do for it~~ `<div class="row">` ~~for it~~

~~<div class="row">~~
~~<div class="col-sm-2 col-lg-3">~~
~~<div class="col-sm-2 col-lg-3">~~

~~after this alternative you don't need to~~
~~for padding about 8px or padding 8px~~
~~as per the~~

we to use this on button with the support of
div
such as
`<div class="row">`
`<div class="col-sm-4 col-lg-6">` fix text is contained
under border at both pixels 576 px or 992 px
button

Javascript

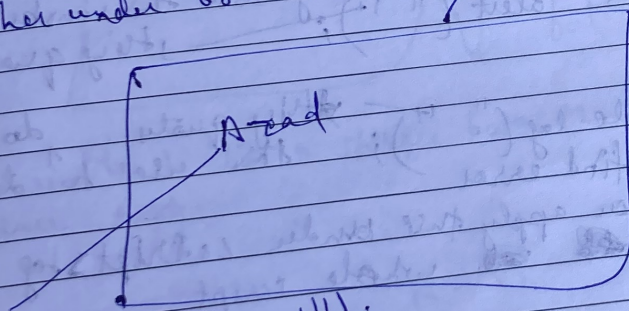
A website where functionalities not working
such as onclick button not submitting or either
working too that's called static website

we turn this static website to dynamic web site
by adding functionalities after added functionalities
that's called dynamic website

Javascript is easy, fast, here everything is
object such as h1, h2 tags here are objects

client side for just viewing the website not to change the server side for just viewing and change the website functionalities too

Javascript uses in html by script tag such as `<script>`/`</script>` tags can be added either in head tag or in body tag
`document.write(" ");` string notations or double quotes or uses to write something under body or either under browser



`document.write("Azed");`
 here Alphabet is a character, a group of character is a string
 A character can be noted by single quotes such as `document.write('A');`

→ This is either a alphabet or a character too
 That's not means single quotes can't work on string and same for character as applying double quotes on it

`window.alert(" ");` ^{string}
 to show something under alert tab
`console.log("hello");` ^{string}
 hello
`alert(" ");` ^{alert tab}

/ browser
 without window keyword this works too such
 as `alert(" ");` ^{string quote}

`console.log(" ");` ^{string quote} ^{do}
 and find error ^{this use to debugging}
 if you apply this under `Script` tab this applied
 on whole script

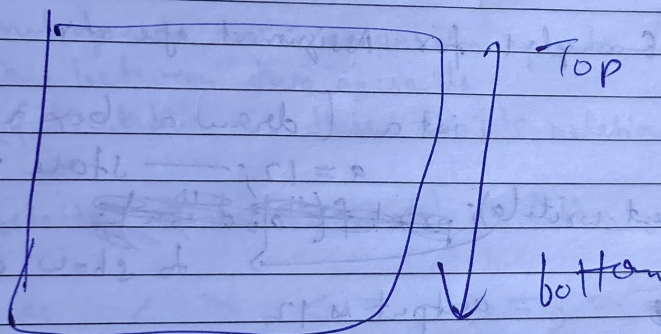
for checking you applied `console.log(" ");`
 go on inspect and click on console and you
 get console window

Operator is to perform something
 operator operates on operand
 such as `c = A * B`
 ↓ ↑
 operand operator
 here c is operand

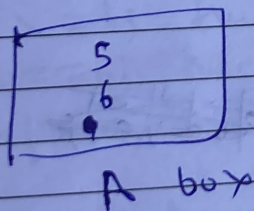
unary operators such as $A++$

binary operators odds two thing
such as $A+B$

Programming happens from top to bottom



Variable is like a storage box that store the value ~~and~~



$A = 5;$
 $A = 6;$
 $A = A + 6;$

A variable has no need of either single quote or double quotes

<script>

document.write("A red");

window.alert("A red") to draw alert window on above body

</script>

Type of operators

- (1) Arithmetic operators
- (2) Assignment operators

Assignment ~~Arithmetic~~ operators such as $=$, $+=$, $-=$, $*=$, $/=$, $\%=$

Arithmetic operators such as $+$, $-$, $*$, $/$, $\%$

Example 1 for Assignment operators

```
int a (draw a box a)
a = 12; // store value in a box
document.write(a); print f("#d", a);
// to show on browser
// output is 12, end body
```

Example 2

```
int a (draw a box a)
a = 12; // store value in a box
a = 3;
a = 34;
document.write(a); print f("#d", a);
// output is 34
```

Example 3

```
int a;
a = 12;
a = a + 4; // or a += 4;
document.write(a); print f("#d", a);
// output is 16
```


Example 4

```
int a;  
a = 12;  
a = 3;  
a = a + 6; // a = 9  
printf("%d", a); document.write(a);  
output is 9
```

Relational operators or comparison operators that tell relations between two operands

if relation is true that's relation return true

if relation is false that's relation return false

such as $<$, $>$, $<=$, $>=$, $==$, $!=$

$a < b$; true or false

$a > b$; true or false

$a <= b$; true or false

$a >= b$; true or false

$a == b$; true or false

$a != b$; true or false

~~Best~~ Logical operator

Relationship between two ~~boolean~~ expression

one ~~boolean~~ expression has either true or false

such as $\&\&$, $\|\|$, $!$

that's may
has either
true or false
logical and

$a < b$ expression 1
 $\&\&$ boolean operator or logical operator
this operator decides ~~exp~~ by $\&\&$ true or false
on decision this also return true or false

this also may
either true or
false
boolean

boolean table
expression 1

T	T	T
T	f	f
f	T	f
f	f	f

expression 2
and decision in behalf
of both expressions

logical operator
expression 1
expression 2

expression 1
a < b
expression 2
a == b

T	T	T
T	f	T
f	T	T
f	f	f

expression 2
or decision in behalf of
both expressions

logical or

increment operator or decrement operator (unary
operator)
such as ++, --

Example 1 (Post-increment)

int a;
a = 12;
a++;

document.write(a);

~~output is 13~~
~~value of a is 13~~
~~value of a is 13~~

Example 2 (Pre-increment)

int a;
a = 12;
++a;

document.write(a);

~~value of a is 13~~
~~value of a is 13~~

output is 12

Example 3 (Post increment)

```
int a, b;  
a = 12;  
b = a++;
```

~~document.write(b);~~ ~~print("val of a is " + a);~~
~~document.write(a);~~ ~~print("val of a is " + a);~~
output is b = 12

a = 13

Example 4 (Pre increment)

```
int a;  
a = 12;  
b = ++a;
```

~~document.write(a);~~ ~~print("val of a is " + a);~~
~~document.write(b);~~ ~~print("val of a is " + a);~~

output is

a = 13

b = 13

In Javascript we use document.write(a);
for variable, document.write("abc"); for string,
document.write('a'); for character
for including output in the above code for
what we performed previous

such as

A = 5;

A = 6;

A = A + 6;

document.write(A);

$a = 5;$
 $a++;$ b (increment by 1)
 $++a;$ b (increment by 1 in a)

$a = b;$

$b = a++;$

↓
 that's assigning operator

that's means b has value of $a++$

$a = 7$ because in second line a is represented
 as $a++$

$b = 7$

$a = b;$

$b = a;$

$a = b$

$b = b$

o/p is modulus that's used in place of
 remainder such as

$$\begin{array}{r} 2 \overline{) 5} \\ \underline{4} \\ 1 \end{array}$$

remainder is 1, modulus is 1

\Rightarrow that's means we equaling the
 things such as $a == b$
 if $a = 2, b = 2$, that's a both to be

equal

This also called equality

Not operator (!)

used for converting false to true or true to false

such as

$a = 6;$

$b = 7;$

$(!(a > b))$

that's condition to be false
but you applied not on that

so finally you got true

OR operator for or operator one of the condition must be true then whole to be ~~also~~ true

such as

$a = 7;$

$b = 6;$

$a \geq b$

||

$b \geq a$

that's true

that's false

so finally you got true

And operator for and operator both of the conditions must be true then whole to be true such as

$a = 7;$

$b = 7;$

$a \geq b$

|| $a = b$

true

true

so finally you got true

$a = 6;$
 $b = 4;$

$a = a + b;$ (increment a by adding a and b)

$a = 10$
 $b = 4$

$a += b;$

(increment a by adding a and b)
 $a = 10$
 $b = 4$

Write a program to find maximum between two numbers

$a = 12;$
 $b = 13;$ (these values may be anything you don't need to worry about) so forget about that after writing

$\text{if}(a > b) \{ \text{document.write}("a \text{ is greater"} + a);$
 $\text{if}(b > a) \{ \text{document.write}(b + "b \text{ is greater}");$

→ There are two numbers so 2 possibilities, one be $a > b$, $b > a$ so we do write both

There is a strategic way that after condition written, you have to write document.write string according to the condition

3
2
1
2
2

such as here saying a greater than b
`if (a > b) { document.write ("a is greater");
 }`

write a program to find maximum between three numbers

if a box has 3 things, that if one is greater than other must be greater than other
 $a(a > b \text{ \& } a > c)$

`a = 12;
 b = 13;
 c = 14;
 if (a > b && a > c) { document.write ("a is greater");
 }`

`if (b > c && b > a) { document.write ("b is greater");
 }`

`if (c > a && c > b) { document.write ("c is greater");
 }`

You can take numbers of if's at the same time note that if you write else after third if means that ~~it~~ belonged to that third if means if else third if to be true then that's else to be false

write a program to check whether a number is negative, positive or zero

that a here belong to negative numbers

```

a = 12;
if (a < 0) { document.write ("a is negative"); }
// not a here belong to positive numbers (above 0)
if (a > 0) { document.write ("a is positive"); }

```

write a program to check whether a number is divisible by 5 and 11 or not

```

a = 11;
if (a % 5 == 0) { document.write ("a is divisible by 5"); }

```

) % means we have first here divide a by 5, then we get remainder a modulus recognised by % sign
for perfect division there must be no remainder
if $a = 5$;

$$\begin{array}{r} 5 \overline{) 5} 1 \\ \underline{5} \\ 0 \end{array}$$

here remainder is zero
so, here a is divisible by 5

for clearly we use $a \% 5 == 0$ for security that that's remainder must be equal to 0

write a program to check whether a number is even or odd

```

a = 9;
if (a % 2 == 0) { document.write ("a is even"); }

```


if a number divisible by 2 that's, be even

if a number divisible by 3 that's, be odd

write a program to check whether a year is leap year or not

$a = 2024;$ have a not divisible by 100 to where we!
 $\text{if}(a \% 4 == 0 \text{ \& \& } a \% 100 != 0)$
 document.write("a is leap year");
 }

~~The year which is divisible by 4 that's a leap year and a year that's not divisible by 100~~

in the case of $!$ sign we don't need to use $==$ double equals as equality we can use single equals too such as $!=$

Writing a string and variable in a same document

$\text{write}(\text{"string"} + (\text{variable}));$

$\text{document.write}(\text{"string"} + a);$ here is variable

output is

for space add space in string such as

$\text{document.write}(\text{"string"} + a);$

output is

string a

here is space

In some cases you have to calculate the things first then apply conditional if on it such as

you want to show calculate as a output for that ~~you~~ do before calculations

`note1 = a/2000;`

`document.write("note1" + note1);`

That's a variable we create for assigning calculations

If you want to update anything value then say this variable name at left side then add this on right side too

`a = a/2000;`

and do whatever you want

So now `a = 1,000` after completion of this `a/2000` calculation

So now if you ~~can~~ create ~~it~~ if then that's belong to updated `a`

`%` modulus uses for gaining remainder or use that too

`|` division uses for gaining quotient or use that too

In some cases we use three step conditional

~~if () {~~
~~then also if~~
~~then also~~
if () { — step 1
}
else if () {
} — step 2
else { — step 3
}

if () to be false then give else it follows ~~it is a correct for us~~

If again else if goes false then else to be followed

Write a program to check whether a character is alphabet or not

```
a = 'A';  
if (a >='A' || a <='Z' || a >='a' || a <='z') {  
    document.write("character is alphabet");  
}  
else {  
    document.write("character is not alphabet");  
}  
}
```

→ This not sufficient to alternative this on next page

Write a program to input angles of a triangle and check whether triangle is valid or not

```
a = 60;  
b = 60;  
c = 60;  
if (a == b && b == c && c == a) {  
    document.write("triangle is valid" + a + b + c);  
}
```


Write a program to check whether character is alphabet or not

```
X = prompt("Enter any input");
if (X >= 'a' && X <= 'z' || X >= 'A' && X <= 'Z')
    document.write("character is alphabet");
}
else {
    document.write("character is not alphabet");
}
```

Write a program to input angles of a triangle and check whether triangle is valid or not

```
a = 60;
b = 60;
c = 60;
if (a == b && b == c && c == a) {
    document.write("triangle is valid" + a + b + c);
}
else {
    document.write("triangle is not valid" + a + b + c);
}
```

Write a program to input angles of a triangle check whether triangle is valid or not

```
a = 40;
b = 60;
c = 80;
if (a + b + c == 180) {
    document.write("triangle is valid");
}
else {
    document.write("triangle is not valid");
}
```


write a program to check whether the triangle is equilateral, isosceles or scalene triangle

```
a = 60;
b = 50;
c = 40;
if (a + b + c == 180) {
    document.write("equilateral triangle");
}
else if (a == b || b == c || c == a) {
    document.write("isosceles triangle");
}
else {
    document.write("scalene triangle");
}
```

Write a program to calculate profit or loss

sp = 500

sale price

pp = 600

purchase price

if (pp > sp) {

loss = pp - sp;

document.write("loss" + loss);

}

if (sp > pp) {

profit = sp - pp;

document.write("profit" + profit);

}

if (sp == pp) {

document.write("no profit or loss");

}

Write a program to find month name by
enter first letter of month
X = prompt("Enter month first letter");
if (X == 'J' || X == 'j') {
document.write("January");

}
if (X == 'F' || X == 'f') {
document.write("February");

}
if (X == 'M' || X == 'm') {
document.write("March");

}
if (X == 'A' || X == 'a') {
document.write("April");

}
if (X == 'M' || X == 'm') {
document.write("May");

}
if (X == 'J' || X == 'j') {
document.write("June");

}
if (X == 'A' || X == 'a') {
document.write("August");

}
if (X == 'S' || X == 's') {
document.write("September");

}
if (X == 'O' || X == 'o') {
document.write("October");

}
if (X == 'N' || X == 'n') {
document.write("November");


```
if (x == 'd' || x == 'D') {
    document.write("December");
}
```

Write a program to input any alphabet
check whether it is vowel or consonant

```
a = 'a';
if (a == 'a' || a == 'e' || a == 'i' || a == 'o' || a == 'u') {
    document.write("vowel");
} else {
    document.write("consonant");
}
```

Write a program to check whether a character
is upper or lowercase

```
a = 'd';
if (a >='a' && a <='z') {
    document.write("lowercase");
}
if (a >='A' && a <='Z') {
    document.write("uppercase");
}
```

Write a program to check whether character is
alphabet, digit or special character

```
a = '@';
if (a >='A' || a <='a') {
    document.write("alphabet");
}
```



```

if (a <= 0 || a >= 10) {
    document.write("digit");
}
if (a == '#' || a == '@' || a == '$') {
    document.write("special character");
}

```

Write a program to find input marks of 5 subjects

```

X = prompt("Enter marks");
if (X >= 450 && X <= 500) {
    b = X / 500 * 100;
    document.write("Grade A" + b);
}

```

```

if (X >= 400 && X < 450) {
    b = X / 500 * 100;
    document.write("Grade B" + b);
}

```

```

if (X >= 350 && X < 400) {
    b = X / 500 * 100;
    document.write("Grade C" + b);
}

```

```

if (X >= 300 && X < 350) {
    b = X / 500 * 100;
    document.write("Grade D" + b);
}

```

```

if (X >= 200 && X < 300) {
    b = X / 500 * 100;
    document.write("Grade E" + b);
}

```

its always
as per the
page

Write a program to find notes in given amount.

```
X = prompt("Enter amount");
if (X >= 2000) {
    note2000 = X / 2000;
    document.write("note 2000 = " + note2000);
    X = X % 2000;
}
if (X >= 1000) {
    note1000 = X / 1000;
    document.write("note 1000 = " + note1000);
}
```

Switch Syntax
 $ch = \rightarrow$ anything placed here either number or character

$switch(ch) \rightarrow$ that's user defined
 That's maybe x, y, z, a, b,
 & such that
 that's predefined

$ch = 's';$ character
 $switch(ch)$
 here assigning user defined value
 // next line
 Comment
 here to create cases

write here to what to show as output
 $document.write("one");$
 $break;$

break is necessary for
 so that it cannot start next case here
 if it is case 2, then 3 and further default

Case 2:

```
{
  document.write("Two");
  break;
}
```

Case 3:

```
{
  document.write("Three");
}
```

Case 5:

```
{
  document.write("Sunday");
  break;
}
```

default:

```
{
  document.write("Invalid choice choice");
  break;
}
```

If you can't create any case then it automatically that default case is executed.

Prompt

```
X = prompt("Enter anything");
```

actually in to write prompt we use thing that's why

after entering anything we get string too
 maybe you get 3 that's be string but
 a number
 enter variable here to convert
 to number
 so sometimes we use parseInt()

such as

~~y~~ = prompt("Enter anything");
 that's variable
 here we enter anything
 may be number
 that's show in a
 y variable

Now convert that it into number, that's a
 string because

~~y~~ = parseInt(~~y~~); here write variable
 to which convert
 that's variable here to update y value

Now i do check whether it is number or
 string?

so i write
 document.write(" " + typeof(X));

that's shows in
 output that x is
 number or string

we use <script> </script>
 object for using

javascript in html
 for using in html we use string quotation is
 'javascript'

if i want to use break in javascript of html

document.write("
" + docof(X));

function

like if you want to do number of tasks then you write code according to that then you want to again do that tasks so what you do it's better to use that code which you first written for that use function

working again and again

function is actually a set of code

Pillars of functions are (1) Body

(2) function calling

that's take time

function body written can't work without function calling

and your project be bigger too

function body

key word

// new line

function functionName ()

code be written here such as document.write("one");

}

function calling

functionName();

such as

```
function a() {  
    document.write("One");  
}  
a();
```

function Parameters these are ~~actually parameters~~
of two types actual parameter and formal
parameters

formal parameter applied to 'Body'
Actual parameter applied to function calling
Both of them applied under () square braces
such as

```
function functionName( ) {  
    // body  
}
```

formal parameter

value write here

```
functionName( );
```

Actual parameter

formal Parameter is actually a variable may be
x, y, z, a, b, c

Actual Parameter is actually a formal parameter
variable value such as 1, 2, 3, One

such as

```
function functionName(x, y) {
    document.write(x+y);
}
```

functionName(1, 2);

output be 3

```
function functionName(x, y) {
    document.write(x);
    document.write(y);
}
```

functionName(1, 2);

output be 12

```
function functionName(x, y) {
    document.write(x);
    document.write(y);
}
```

functionName(1, 2);

Output be 11

Write a program to find input marks of 5 subjects

```
X = prompt("Enter Marks");
b = x / 500 * 100;
if (b >= 90 && b <= 100) {
    document.write("Grade A" + b);
}
```

```
if (b >= 80 && b < 90) {
    document.write("Grade B" + b);
}
```

```
if (b >= 70 && b < 80) {
    document.write("Grade C" + b);
}
```

```
if (b >= 60 && b < 70) {
    document.write("Grade D" + b);
}
```

```
if (b >= 40 && b < 60) {
    document.write("Grade E" + b);
}
```

function return it's just used after successful done of tasks after calling it's just for using to do a return or message that that's tasks are completely done

That's used in body after the code which you already written function

That's actually replaces function name by return value

Such


```
function functionName(x,y)
{
    document.write("One");
}
```

functionName(1,2);

So Now a task is done

And if a function returns

```
function functionName(x,y)
{
    document.write("One");
    return "yes";
}
```

```
x = functionName(1,2);
document.write(x);
```

A function has one return such as

```
function a(x,y)
{
    if(x%2==0){
        return "Even";
    }
    if(y%2==0){
        return "Even";
    }
    if(x%3==0){
        return "Odd";
    }
}
```


if ($y \% 3 == 0$)
return "odd";

↗ here x is odd, y is even

x = a(3, 2);
document.write (x);

conditions with $x \% 2 == 0$
, $y \% 3 == 0$ to be true

~~if there is output is first odd~~
~~even in y as return then odd is~~
~~as return~~
~~but a function has one return~~

~~if you get one return then other return~~
~~is not executed~~

so there are two returns and shown as output doo
it's automatic that programming do do do do
first return to be chosen first as programming
goes from top to bottom and the other return
to be remain same

But if you ~~commented~~ commented one already
chosen return then ~~the~~ return goes on work
and shown as output rather other return first

you can check the return by document.write
(x);



DOM (Document Object Model)

That means
get the element or either object
if you got that object in console you got
not object
if not you got that object in console you
got null

What is document as here such as

`<h1> This is Rohit </h1>`

That's whole a document

The written thing between the `<h1>` `</h1>`
tags this is innerHTML.

if you want to update innerHTML which you
already wrote then
write first here variable (here x)

This has document.`getElementById("h1 tag");`
that means x has h1 tag string stored, now you
want to update that

so `x.innerHTML = "` " "

want to update

write here what you

DOM works is to get the document
element (such as h1 tag and more), by specified
Id, id is actually written on the object or
element to whom which to get

such as
in HTML

document

```
<body>  
  <div id="div">  
  </div>  
</body>
```

want to get the div first element but first is document.

this id created here

Now Javascript part

```
<script>  
x = document.getElementById("div");  
you can use you can use console.log(x);  
or may use document.write(x);  
use for check what has x stored
```

```
</script>
```

Now you want to update innerHTML

```
x.innerHTML = "OK";
```

string write here do which update here it is OK

onclick event is used to call the function as per we need it generally linked to button tag

```
<button onclick="function calling here">button name here</button>
```


~~script~~
function abc()
↓
x=document.getElementById("div1");
console.log(x);
x.innerHTML="OK";
~~document.getElementById("div1").innerHTML="OK";~~
} ~~script~~
function body

function calling ~~on window~~ ~~body~~ ~~body~~
here is conditional

<button onclick="abc()" >button text here / </button>

here onclick function to be call
<body> <div id="div1"> here it is robot </div> </body>

~~the basically focus on first element function~~
~~body~~

window ~~document~~ is global object
for gaining other global objects in console only
use keyword window

x=window;
~~document~~ ~~document~~ console.log(x);
on console, tap ↗ window for gaining list of
global object
vision

Selection statements are of three types

- ① if
- ② if-else
- ③ else-if

Loop is continuous repetition of ^{some} code that's automatic by conditional operator such as greater than, less than or equal to

Loop of three types

- ① for loop
- ② while loop
- ③ do while loop

step 3 goes on

Three of both work on same steps such as

For loop

its code written in one line

Syntax for (initialization, condition, increment/decrement)
statement to be written here

such as

```
for (i=1; i<=10; i++)
```

```
document.write("One");
```

```
}
```

- ① initialization
- ② condition (if true then)
- ③ Statement execution which to be written under condition
- ④ Increment / decrement
- ⑤ Again check condition
- ⑥ continuous goes on

while loop its code written in column such as

Syntax ~~while~~ initialization, while (condition here)

✓ Statement to be written here:
// new line Increment | Decrement
}

for and while ^{loop} almost same

do while loop
syntax initialization;
keyword do {
 statement to be written here
 // new line Increment | Decrement

keyword while (condition here);

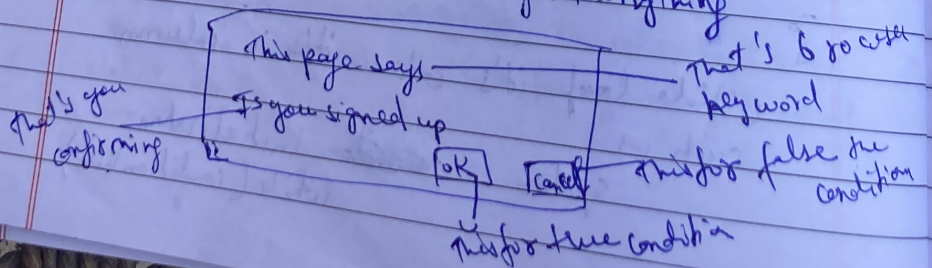
break and continue

Actually ~~break~~ break keyword is used to stop the loop at specific condition

Continue is to skip that specific condition to whom continue keyword is applied

confirm box and Alert Box

confirm box is used to ^{do} confirm anything



sting written here

Green House
Date _____
Page No. _____

$x = \text{confirm}(" \wedge ");$ here it is

Is you signed up

confirm box based on true or false so
conditional if else must be used

if (x) Is you signed up

when user click ok then x value
matches the user defined value so if to
be true

if (x)

document.write("Is you signed");

}
else

{
document.write("You're not signed up yet");
}

for loop

for (i=1; i<=10; i++)

document.write("One" + i + "
");

output is

One 1

upto

One 10

Now ~~if~~ you want to break the loop
at $i=5$ then

Superbrand
INDIA
Consumer Validated
2019

Classmate - India's No. 1*
notebook brand is now a Superbrand.
*Survey conducted by IMRB in Jan 2019.

Now write the if ~~as~~ as statement
then use break keyword in next line

```
for(i=1; i<=10; i++)  
{  
    if(i==5)  
    {  
        break;  
}
```

```
break;  
break;  
break;  
break;  
break;
```

If you want to skip the loop at some specific condition

```
i = 4  
for(i=1; i<=10; i++) {  
    document.write("One " + i + "<br>");  
    if(i==4)
```

one
one
one
one
one
one
one
one
one
one

continue;

```
}  
    continue;
```

document.write("One " + i + "
");
In break and continue

document.write("Skip here");
Keyword continue and break

with break keyword

```
for (i=1; i<=10; i++) {  
    if (i==5) {  
        break;
```

```
    }  
    document.write("One" + i + "<br>");
```

→ that means we taking the output after using the keyword break

output is

One1

One2

One3

One4

while loop

```
i=1;  
while (i<=10) {  
    document.write("One" + i + "<br>");  
    i++;  
}
```

output is

One1

One2

One3

while loop with break keyword

```
i = 1;
while (i <= 10)
{
    if (i == 5)
        break;
```

```
    document.write("One" + i + "<br>");
    i++;
}
```

Output is

```
One
One
One
One
One
```

do while loop

```
i = 1;
do {
    document.write("One" + i + "<br>");
    i++;
```

```
while (i <= 10);
```

Output is

```
One
One
One
One
One
```

do while loop with break keyword


```

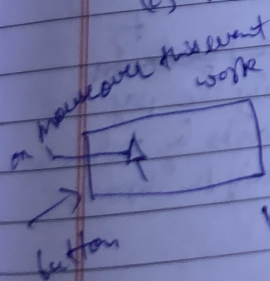
i = 1;
do {
    if (i == 5)
        break;
    document.write("One" + i + "<br>");
    i++;
} while (i <= 6);
    
```

Events

- (1) Click Events
- (2) Key Events
- (3) Mouse Events
- (4) (1) onfocus event
(2) onblur event

- (1) Click Events
- (1) onclick → on single click this event work
 - (2) ondblclick → on doubleclick this event work
 - (3) oncontextmenu → on rightclick this event work

- (2) Mouse Events
- (1) onmouseover
 - (2) onmouseout
 - (3) onmousedown → on single click this event work
 - (4) onmouseup



- (3) Key Events
- (1) onkeydown
 - (2) onkeyup
 - (3) onkeypress

on mousedown this event work but first we do mouseup then his works

on mouseup this event work but first we do mousedown but when we left mouseup from click position this event to be work

these events are actually works with input tag

(3) Key Events

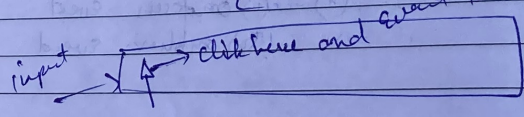
- (1) onkeydown → on key press this works
→ keys of all types (Alphabet, number, special character, functional keys such as ctrl, Alt and more)
- (2) onkeyup
- (3) onkeypress → on key press this event works with (Alphabets, number, Special character but not with functional keys)

This event works with first we do page layout but we left one key from click position this event do we work

(4) (1) onfocus Event

→ This works with when you do focus on something in input by clicking inside input tag

These events works with input



This event work with when you do first onfocus Event and then click outside the input that's a blur Event

(5) Load Events

These events are work with body

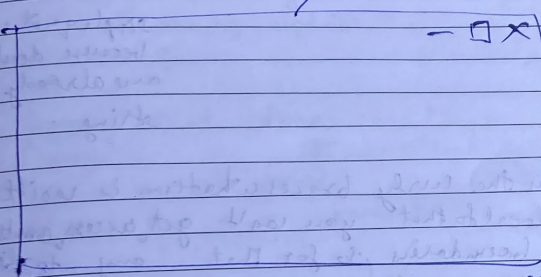
- (1) onload Event
- (2) onresize Event
- (3) onunload Event

→ on resizing the page this event works by when we do resizing the browser size

on successful page load this event works

Browser window

Browser window



such as `h1` tag

style the ~~valid~~ ~~string~~ body tag element
if we ~~not~~ ~~one~~ ~~in~~ `<body>`
~~id="h1"~~ `<h1>` `</h1>`

Now style that
~~id="h1"~~ `id="h1"`
`<script>` `</body>`
`X = document.getElementById("h1");`
`X.style.color = "Red";`

~~background~~ `X.style.background = "green";` `</script>`
↓
not for background color input tag id
That must body tag

Generally input has attribute value its manually
be updated by user when use input anything
but we can do it by default too by
`X = document.getElementById("Input here");`
`X.value = "Submitted";`

Update the onclick event ~~page~~ ~~hyper~~ ~~text~~ `hyper` text
after successful onclick or single click
That's means we updating the `hyper` text through
onclick event itself for this onclick
we use this keyword
or not by other variables

button onclick="this.innerHTML='Signup'"

single quotes here
because double quotes
are already used as
string.

under the curly braces whatever is written that
is local to that you can't get access outside
these boundaries, so for that any to element which
you have to first get that element into
your local or under curly braces then
you can update that

id keyword here

```

<
  x = document.getElementById("id");
  x.innerHTML = "One";
}

```

Datatypes

- (1) Number [all types of numbers such as
real numbers, irrational numbers and more]
- (2) String → ' ' single quotes, " " double quotes,
" " " triple quotes
if you writing another string under one string
then use single quotes for that

- (3) Boolean → True, false
- (4) Null → Null can't accept 0
x = null then you can't update x too
- (5) undefined → x = undefined that's mean

That you not get defined x
may be later you can do define that
such as by update the x by some
value

(6) Array that stores multiple elements
such as $x = [1, 2]$ element₁
element₂ here is two elements

elements are stored by index, index starts from 0
syntax variable here = $[1, 2, 3, 4, 5, 6]$ ^{if you} _{bracket}
index _{0 1 2 3 4 5} then are six elements

Number of elements defined array size so here
array size is 6

index find by this formula, $0 = n - 1$ ^{index}

here element 1 stored at 0 index ^{number of} _{elements}

if we do $x[0]$

then we get 1, that is element which
is stored at index 0 we checking here element
That's means ~~at~~ ^{at} index 0 and that's do
be stored in variable x here

2) Object → That's cover key/property and their value
syntax →

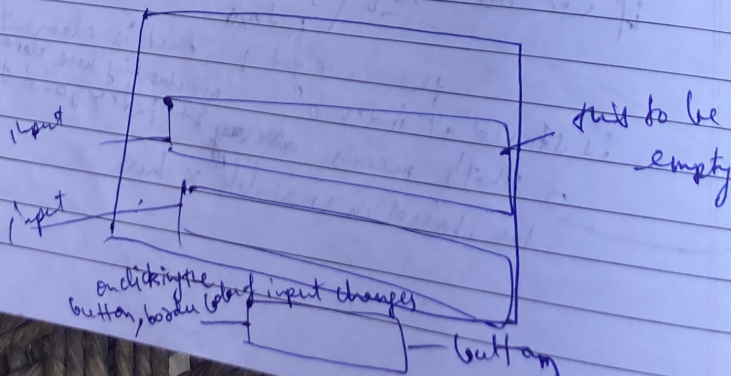
Variable here = {
key/property here : value
name : "Rohit",
class : "abc",
Roll No : "xy2",
}

you can write here number too

if you want to check name property then

variable here . name
that's give you name value at console

Validation works on input tag when you not typed anything input border to be turn red that's ensure validation that's means if user not typed yet, and trying to login
username and password inside input tags




```

id = "input1"
<body>
  <button onclick = "abc()" > click me
  <input type = "text" placeholder = " " />
  <script>
    function abc()
    {
      x = document.getElementById("input1");
      y = document.getElementById("input2");
      if (x.value == " ")
      {
        id = "input2"
        <input type = "password" placeholder = "password" />
        <p id = "paragraph" > </p>
      }
    }
  </script>

```

```

x.style.border = "2px red solid;";
y.style.border = "2px red solid;";
z.style.color = "red!";
z.innerHTML = "You not write anything yet!";

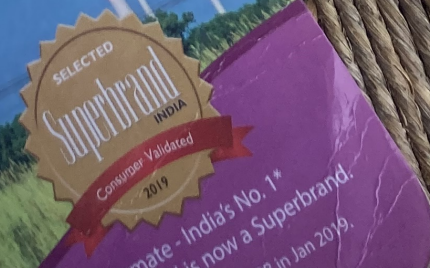
```

```

} else {
  x.style.border = "2px green solid;";
  y.style.border = "2px green solid;";
}

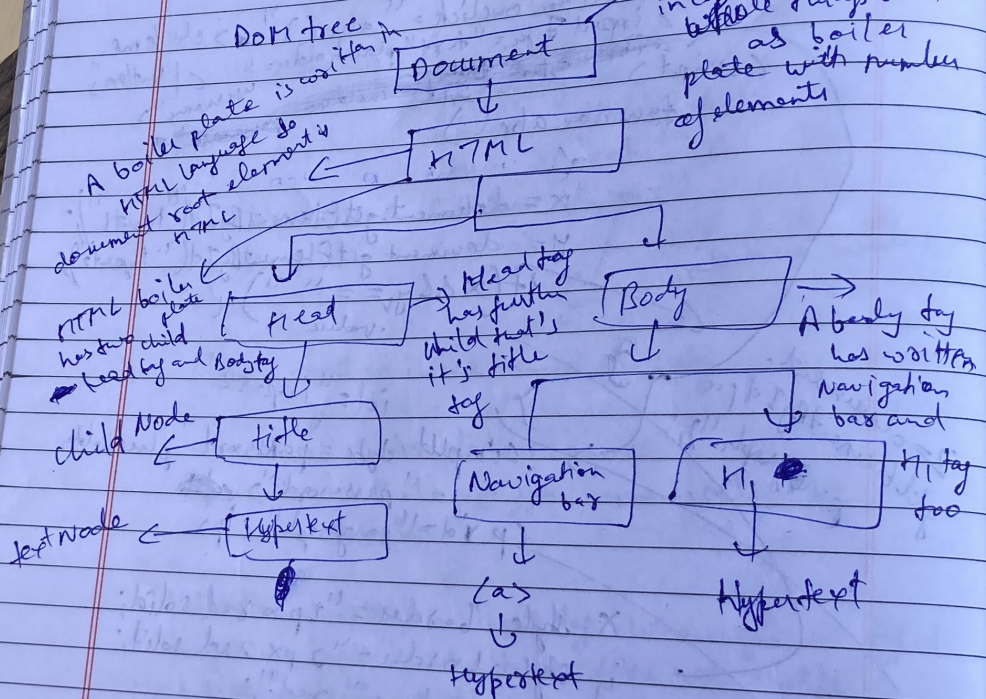
```

In the final we set the paragraph `innerHTML` to be empty or not write yet. This can be update by `innerHTML` in javascript.



DOM (Document Object Model) is advance

This is complete file in which we write boiler things such as boiler plate with number of elements



~~console.log~~ console.log(document);
we are targeting here document
on targeting this we get complete boiler plate which you written with elements or tags

boiler plate or a complete document

```

<!DOCTYPE html>
<html>
  <head>
    <title>
    </title>
  </head>
  <body>
    <h1>
    </h1>
  </body>
</html>
  
```


Target Types

- (1) Target By Id
- (2) Target By class
- (3) Target By Tag Name

(1) Target By Id

Id is unique ~~and~~, so that's unique for every element. Targeting means we getting the target thing here that's element

Targetting or getting the element by Id
document.getElementById("Id be written here");

(2) Target By class

Here it's not unique id, here is class ~~not~~
one class may linked to number of ~~class~~ elements
so we use here term called elements

Targetting or getting the elements by class
document.getElementsByClassName("class name here");
↓ that's standard

(3) ~~Targeting~~ Target By Tag Name

Here is the case of Tag Name
one tag ~~may~~ may used number of time
so we use here called elements too

Targetting or getting the elements by Tag Name
document.getElementsByTagName("Tag Name written here");



Classmate - India's No. 1*
brand is now a Superbrand.
by IMRB in Jan 2019.

console.log (document.all);
That's give you boiler plate or document tags in a order way in you created

This (document.all) give your document in a ^{key names} tags ~~order way~~ ~~that's not give you child tags such as head body child style etc~~

So basically it's give a overview of tags ^{by key names} such as html, head, body

document.all give ~~that key~~ document tags key names in a array form
this keyword for taking document output with document.all every square brackets

HTMLAllCollection(8) (html, head, meta, title, ~~script~~)
index 0: html
index 1: head
index 2: meta
This saying how much number of ~~document~~ elements used

~~key word~~
~~says html language~~
~~how much~~
~~collection of tag~~
~~tags or that's~~
~~creating all~~
~~elements which~~
~~document is~~
~~that~~

```
<html>  
<head>  
<meta>  
<title>  
<script>
```


Variable stored in an array at the different indexes

so we get ~~key~~ index of the elements too by writing the `console.log(document.all)`

`console.log(document.all[5]);`

we getting here key tag which is stored at index 5

`console.log(document.links);`

key words may be link, links, baseURI or more

we getting here links if anyone added if not you created them you get

`document.links`

work ~~on~~

mostly ~~if~~ if you have one link or more

HTMLCollection

length: 0

this works only with keys of multiple links

if you added `HTMLCollection(2)`

baseURI has type (1) link

index 0: a: anchor (9, 9)

length: 2

`console.log(document.baseURI);`

gives you

right now webpage html path such as

`http://127.0.0.1:5501/login.html`

`console.log(document.link);`

that's give you undefined because you not created link here

`console.log(document.location);`

give you link

origin, pathname

such as `"/login.html"`



India's No. 1 Superbrand

</body>

<script>

x = document.getElementsByClassName("div");
console.log(x);

<div class="div">

</div> <div class="div">
</div>

</body>

Output is HTMLCollection []

~~at index 0~~

index 0 → 0 = div-div
div element that element has linked class name that is div

If you stored another get element in same variable in which you previously done such as here is x work

not's previous work done

x = document.getElementsByClassName("div");
now stored another element in that variable

x = document.getElementsByTagName("div");
output is

HTMLCollection []

0 = div-div

1 = div-div

that's div tag
that's class linked
that's class linked too
length: 2 (because here two elements or tags)

React.js

- (1) Library of Javascript
- (2) Open source (you don't need to buy subscriptions to use them or paid them)
- (3) Used for create UI (user Interface) of web application
It's just for viewing to customers not properly work at all.
- (4) Approach it's to just view or for visual the website and just other functions to be not working at all
- (5) frontend language

Library React.js

has inbuilt functions
It has view (or v)
It just direct way to view

framework

Angular is framework of Javascript
has inbuilt functions
It has MVC (Module view controller) such as process starts from module to then controller to then view

NPX

It's be used in particular project which you working on

NPM

It's be used Globally by CMD (command)

It's a Node Package Executor or Runner

It's a Node Packages manager



Classmate - India's No. 1*
notebook brand is now a Superbrand.
conducted by IMRB in Jan 2019.

NPM for
executing the
downloaded
packages (by NPM)

NPM for download or
install the packages

for proper functioning of the React.js
you must install Node.js in your
device

after successfully installed check that install or
not by
open command and type `node -v`

for node version

for successful run of the React.js projects
must install NPM globally

`npm install -g create-react-app`
global

after successfully installed check that install
or not by
open command and type `npm -v`

for npm version

Newest way to open project folder
Right click that folder and click open in
terminal and type `code .` & dots and then
enter, you get visual studio code to be
open

Now click terminal and type `npm create-react-app`
"projectNamehere"

This will create a project to the specified
path or folder
React.js

`cd` is the shortcut key for move

such

`cd desktop` (used for move to
desktop)

`npm start` is the abbreviation for start or
run the project which you right created or
view that at the local host

React.js is a single page application because
there is ~~not~~ need of go live or ~~need~~ of
refresh the page but still if you modifying
the sources changes so be seen at the view
or at the web application

If you downloaded packages or install one that's
done under the `node_modules` folder in your
project

so do of the work we done in the App.js
in the react.js projects

Project dependency based on `node_modules` folder
that's contain resources used for your react project
for successful run



Classmate - India's No. 1*
book brand is now a Superbrand.
Certified by IMRB in Jan 2019.

we first ^{install} "npm install -g" by keyword here "↓"

create-react-app

we may use this keyword to access
installed or downloaded packages
during project

Then we ~~use~~ use npm packages to particular
project

npm "npm keyword here" "here project name"

keyword
for accessing or
running the
npm packages
at your
own project

create-react-app

npm install "Package name" here
install that package in your node-modules

DOM in advance

DOM GET

- (1) innerText
- (2) innerHTML
- (3) GetAttribute
- (4) Attributes
- (5) GetAttribute Node

DOM SET

- (1) innerText
- (2) innerHTML
- (3) Set Attribute

DOM GET and SET

- (1) innerText

<div>

h1 id="h1">hello</h1>

</div>

text between tags

↑
<h1>innerText
</h1>

↑
this shows innerText of the element (target)
↑ By id specific

<script>

x = document.getElementById("h1").innerText = "hello";

↑
updated text written here

↓
whole it's a innerText SET

console.log(x);
</script>

(2) innerHTML

<div>

h1 id="h1">hello</h1>

</div>



Classmate - India's No. 1*
book brand is now a Superbrand.
Validated by IMRB in Jan 2019.

In console you get value of that attribute

Set Attribute

```
<div>
  <div class="class" id="id" name="name" style="border: 2px solid red;">
</div>
```

<script>

x=document.getElementById("Id be written here");
after setAttribute we must to getAttribute again

```
x=document.getElementById("attribute name written here");  
console.log(x);
```

Get Attribute Node

Get Attribute gives you a attribute value
But by Get Attribute Node, you get attribute with value

```
<div>
  <div class="class" id="id" name="name" style="border: 2px solid red;">
</div>
```

<script>

```
x=document.getElementById("Id be written here");  
getAttributeNode("attribute name be written here");
```

</script>



Classmate - India's No. 1*
notebook brand is now a Superbrand.
*Survey conducted by IMRB in Jan 2019.

output is
name = "name1"

To check how many numbers of attributes
linked to your ~~for~~ element

<div>
<h1 id="h1" name="name1">hello </h1>
</div>
This may be to get numbers of attributes

<script>
x=document.getElementById("h1").attributes;
console.log(x);
</script>

its a keyword to get
numbers of attributes
to your element

output you get in array form

```

NameNodeMap(0: id, 1: name, id: id, name: name,
0: id                                length: 2)
1: name
id: id
name: name
length: 2
    
```

after get attributes we can get value to
the specific attribute by ~~the~~ index of array
name attribute at index 1

<script>

x=document.getElementById("h1").attributes

[]
array

you get

name: "name"

index of element
to whom you
can get be
with here
attribute and its

Remove attribute, that remove the value of too
~~for attribute~~ to whom this
applied

<div>

h1 id="h1" name="name">hello</h1>

</div>

<script>

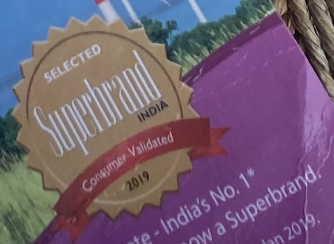
x=document.getElementById("h1").removeAttribute
("attribute name
below the here");

console.log(x);

so you get undefined as output on console

CSS selector are of three types you already
studied in CSS mostly of course in internal CSS

(1) ~~Id~~ selector -> put # hash sign under



Syntax

(i) class selector \rightarrow part ^{dot} under style tag, that's result about class selector

Syntax

(3) Tag selectin \rightarrow put tag keyword under style tag,
that's recall about tag selection
syntax

Syntax

Its alternative
query selector is document

document.getElementById("id to be written here")

query selector only is to get the whole element.

querySelector with the usage of tag selector

```
x = document.querySelector("h1")
console.log(x);
```

tag selector be written here with prefix # sign such as

#h1

This give you whole element as output by which you applied Id by some value

Id value linked to element by whom we accessing here

che id="h1" name="name1" style="border: 2px solid red;">hello</h1>
In output you get that Id with value too

querySelector with the usage of class selector

```
x = document.querySelector(".h1")
console.log(x);
```

class selector be written here with prefix . such as

.h1 or class value

class name, this applied to element by which

output 2

che id="h1" name="name1" class="h1" style="border: 2px solid red;">hello</h1>
output same as tag selector

case



querySelector with the usage of tag selector

`x = document.querySelector("h1");`
tag name or keyword
be written here
sent as `h1`

`console.log(x);`

and put it is

```
<h1 id="h1" name="hame1" class="h1" style="border: 2px red solid">hello</h1>
```

querySelector has automatic first element select property

such

```
<h1>hello</h1>
```

```
<h1>hello</h1>
```

there are two `h1` tags

but when we select by tag selector and get ~~any~~ element by `querySelector` you get only first element ^{whole} not second one

output is

```
h1>hello</h1>
```


if you take one element under ~~the~~ other element by same ~~element~~ tag

`<h1>`
`<h1>hello</h1>`

`</h1>`
output is `<h1>` ~~`<h1>hello</h1>`~~
`</h1>`

only first element
~~selected element~~
element
even that that element has other element

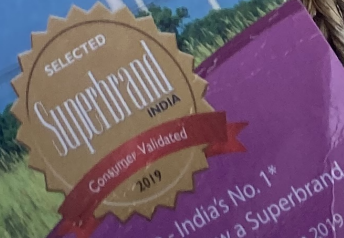
for selecting ~~all~~ all element by same name or keyword we use `querySelectorAll`

`<script>`
`x=document.querySelector("x")`

if you tag selector then tag name or keyword be written here " ;
with # hash sign

`console.log(x);`

`</script>`



first tag
second tag
`<div>`
`<h1 id="h1" name="hanel" class="h1" style="border: 2px solid green; color: red; text-align: center;">hello3</h1>`
`</div>`

output in the form of array
`document.getElementsByTagName("h1")`
`NodeList(3)` [h1, h1#h1.h1, h1]
 there are 3 elements
 element with id included
 and class included too
 0: h1
 1: h1#h1.h1
 2: h1
 length: 3

If you want to get element at specific index
 h1 element with index 2
`document.getElementsByTagName("h1")[2]`
 console.log(x);
 this is array index

you got index 2 element
`<h1>hello3</h1>`

style of types

① normal style

document.querySelector(" ").style.color
tag selector use if
you want be written
such as h1 with prefix
hash sign for
id selector

attribute for styling
property
style = "color"
property value
be written here

output is
hello with red color style

② Capitalised style

document.querySelector(" ").style.backgroundColor
tag selector use if
you want such as
h1 with prefix
hash sign for
id selector

here its red with
original in CSS is
background-color but
we use
background-color here
property value
be written here
here its green

output is
hypertext with green background color

for getting className or value by id linked to
that element through querySelector

x = document.querySelector(" ").className; class value
tag selector use here if want
to want to use

console.log(x);



outputs & h,

manipulate means update something

NPM

NPX

~~NPX is not a package~~

we borrow something from
npm its for limited time
for single project
such as create-react-
app package and
npv execute that too

~~create-react-app~~

NPM downloads the create-react-
package -app

classList keyword not says about multiple ~~class~~ ~~names~~

it's just give you className or classvalue in
array form

<h1 class="h1">

</h1>

keyword to get
className
in array
form

(script)

x = document.querySelector("h1")
tag selector

written here

console.log(x);

how number of class
name

const p =

DOMTokenList({})

linked to that

0: "h1" className

length: 1

value: "h1"

for getting two class at classList use
`<h1 class="h1 h2">` ~~or~~ `<h1 class="h1 h2">`
`document.querySelector("h1").classList`
`console.log(x);`
if you want to get classList in array form
to about specific index

`document.querySelector("h1").classList`
tag selector be written here

`console.log(x);`

index number to
be written here

Add className under classList

`<script>`
`document.querySelector("h1").classList.add("h1")`

tag selector be
written here

className to
which you
want to add

to whom you
add class to
not or
applied that
class properties
to that

`</script>`

`<head>`

`<style>`

`h1 {`

`background-color: blue;`

`}`

`</style>`

`</head>`

querySelector add className be written here we trying first that class then
Remove className under classList add Remove that
same class

`<script>`

`document.querySelector("h1").classList.remove("h1")`

tag selector be written here
to whom you remove class to

className be
written to whom
which to be remove

`</script>`

~~output is comes with~~

callback function

→ $x = \text{setTimeout}(\text{function-name}, \text{Time in milisee} \text{ - only})$
1s = 1000 milliseconds or ms

$\text{clearTimeout}()$

(2) $\text{setTimeout}(\text{function-name}, \text{Time in ms});$
 $\text{clearInterval}()$

callback function is used to call the function on specific time such as after 3 seconds, so that's based on time property here we use setTimeout keyword for that for setting of specific time

callback function is once run only on single refresh if you want to stop the ~~the~~ setTimeout then we do use other keyword called clearTimeout or clearInterval callback function stopped at specific of time where we called such as setTimeout is goes on but sudden we use the clearTimeout at time = 3s then on setTimeout arrives time = 3s callback function stopped on

for repetition of setTimeout we may use other keyword called setInterval

for stop this interval we may use clearInterval
keyword

callback function with setTimeout
that's actual parameter for setting time for
keyword call the required function

(script) {

x = setTimeout(functionNameHere, timeIn)

→ this variable (that's ms);
but it is abc 3000ms

(script) use ~~for~~ for the 3000 ms

function abc() {
clearTimeout or this variable
has stored
setTimeout

alert("successful");

function body

So, function to be call after 3 sec

callback function with clearTimeout

variable comes here

so we first setTimeout then we do
clearTimeout

create a new function body
function abc1()

here we use clearTimeout (x);
alert("cancel");
here it is x
actual parameter
which is set
at

now we to call this function here this function is
abc1

we can do that by button tag in html

`<body>`
`<button onclick="abc1()">Click </button>`
call here
abc1 function

`</body>`

Now will stop all the `setTimeout`
for repeating the callback or ~~function~~ to do
repeating the function till we do stop
even one refresh the page

callback function with `setInterval`
`<script>` keyword for repeatedly calling the function
again and again
`var x = setInterval(function-name here, Time (in ms));`
`function abc3() {`
`alert("1");`
`}`
here it is 3000
abc3
↓
not stop
3s or
3000ms

callback function with `clearInterval`

so we `setInterval` then we do `clearInterval`

Create a new function body

`function abc4() {`
`clearInterval(x);`
`alert("stop");`
`}`
formal parameter

isolate
comes here

<body>

<button onclick = "abc()">stop </button>

</body>

so when we click stop button this will stop the setInterval

Advance usage of setInterval

① setInterval with styling
generally it is obvious that we do first use setInterval but we can have setInterval

(script)
setInterval(functionName, time in ms);

↓
abc

↓ 3000

function abc()

document.querySelector("v") .style.backgroundColor = "white";

}

class selector
here with prefix such as .class

</script>

<body>

<div class = "class">
</div>

</body>


```
<head>
<style> .class {
  border: 2px solid red;
  width: 500px;
  height: 50px;
  background-color: green;
}
</head>
</style>
```

That is, mean after 3 sec
background color goes white
class with setInterval

Calculating and implementing in styling properties
such as width applies according to calculation

for this accept any variable
 $x = 0;$ // x variable assign initially to zero

Calculation, x variable increases with number to
 $x = x + 10;$

So, the width = $x + "0\%$;
for showing the value of x with percentage sign

setInterval (function-name here, Time in ms);
this for once initialise $x = 0;$ abcs 3000

repetition part function abcs() because that's part only calculation here part of repetition

```
document.querySelector(".class").style.backgroundColor = "white";
document.querySelector(".class").style.width =  $x + "0\%$ ;
```


actually $x = x(0)$
takes values first 10, then 20, 30
~~up to 100~~
and goes on

for showing the values of properties such as
margin are px (pixels), width (in px)
height (in pixels)

setInterval with the usage of styling, calculations
and conditional if

(script)
setInterval (function-name here, Time in ms);

initialise here $x = 0$;
function abc(s) {
 ← calculation here

$x = x + 10$;

document.querySelector(".class").style.backgroundColor = "white";

document.querySelector(".class").style.width = $x + "0\%$;

if ($x == 50$) {
 document.querySelector(".class").innerText = "50%
 space filled up";

document.querySelector(".class").style.backgroundColor = "blue";
}

if ($x == 90$) {

document.querySelector(".class").style.backgroundColor = "red";
document.querySelector(".class").innerText = "space is almost
filled up";

}


```

if (x > 90)
  document.querySelector(".class").style.marginTop = x + "px";
}
</script>

```

React


here are components such as Navigation bar, we write code in components because by that we can easily change that code accordingly to the specified type ~~code~~, that fast the project done time

Props

In components we create a component called form component then we write information in this form and that information is called props then we send that or forward that form to other place.

React single webpage application as this needs single refresh only then don't need to refresh again ~~even~~ still the content refreshes automatically


React necessary extensions

- (1) Thunder client
open VS code and click this icon  and search thunder client
This basically used for check the api
- (2) ~~ES7~~ React/Redux/GraphQL/React-Native snippets
this used for fast writing the code
- (3) Bracket Pair Colorization Tagger
used for Brackets colorization such as

{ }
[]
()
red color

{ }
[]
()
blue color
- (4) Auto Rename Tag
used for whenever we change the tag name then its ending tag automatically changed
- (5) Prettier - Code formatter

React 90% of the code written in App file
due to React cause we break this App file
into number of components such as complete App.js
for javascript based App file, App.css for CSS based
App file

for use CSS under App.js file we do  import that
file in our App.js file by `import './file name'`
By default project accepts `src` keyword as folder name its `App.js` single quotes file name
written here
says under folder with
extension



In Javascript we read that function body must need function call for to do all the tasks

Here function body no need of function call here we use return type keyword for showing javascript code to output

we write return type keyword as part of function body

```
function App() {  
  return ( // this code written for  
    <div> // showing at output  
    </div>  
  );  
}  
export default App;
```

return type keyword

html

is that html code output shows at output

this to be study further

In CSS we use class keyword for showing that is class with some name

But in react we use className keyword

React components are of two types

(1) function based components
such as After import line

We write function body code then it is function based component

(2) Class based component
such as After import line we write ~~class~~
~~class~~ class Name then curly braces such as
App.js

class Name

If we create a another div under function that's create a error for that create another div and take all code whatever written under function body and error got solved

JSX

In original javascript we need to write html code as under string ~~code~~

But JSX allows there is no ^{need} of write html code under string, we can directly write html code ~~code~~

for successful ^{run} of React project by npm start, you must in that folder which contains files such as node_modules, src, public

Javascript in Advance

Variable types or notations or create variable concept notation

- Variable 1 (1) let keyword
(2) var keyword
(3) const



variables such as x, y, z or a, b, c can easily store assigned values and you get output on that variable

but in same way we have modernised the way to write variable such as use keywords such as let, var, const before write the variable

~~that is not the way~~
var keyword with assigned value, if you use ~~var~~ same variable and assigned value to that, then you get second value is other shown or overwrite, even vs code don't give you error on this too

<script>

~~var a = 12;~~
~~var a = 13;~~ output is 130
</script> document.write(a);

Variable keyword

let keyword

<script>

let a = 12;

let a = 13;

error you already declared the variable
document.write(a);

</script>

output is blank page or check error at console

So in this let is advanced version of

var

const keyword has specific meaning as per particular usage such as the code is written by specific author and other's so can't change this condition, we use const keyword

<script>

const a = "seena";
document.write(a);

output & seema

</script>

if we write anything to same variable, the output be broken

<script>

const a = "seena";
a = 12;
document.write(a);

output &
blank space

</script>

so const keyword is used once on one variable and then assigned value to that variable

if you write or assigned different value to that same variable in a new line, output be broken

Right way to use var keyword

var a = 12;

var a = 13;

same variable but two var keyword not, wrong

right way

var a; // create variable by var keyword
a = 12; // initialize value to that variable

Const

const can't relate to both of let and var key word

var

```
var a = 12;  
var a = 13;
```

(1) you not get error of same variable assigning different values

(2) output is overwrite of a second value assigned

(3) Here redeclare possible and assigning that to same variable possible too such as

```
let a = 12;  
var a = 13;
```

output is 13

document.write(a);

```
var a;
```

```
a = 1;
```

```
a = 2;
```

output is 2

document.write(a);

let

```
let a = 12;  
let a = 13;
```

(1) you get error of same variable assigning different values

(2) output is broken here due to error comes

(3) Here redeclare of same variable not possible but assigning to that same variable is possible such as

```
let a;
```

```
a = 1;
```

```
a = 2;
```


document.write

```
(a);  
output is 2
```



React

Run react project output onto your phone or device, by default where you write code, that system has access to that project by local host ip such as for react vs code local host ip is `localhost:3000`, if you want to watch `http://` this ip on your phone, your phone connected to that wifi, which your code will then device connected, then by type this ip on your phone and you got that project output

Ctrl +

when you want to search anything, use that shortcut but you ~~click~~ click > arrow sign you'll get replace tab in which you have to write replaced keyword such as you search for excel but you want to change that into axis, for that type axis in replace tab and then click this icon  for successful change

In href don't use # sign that's irrelevant in react, for that you must replace that by / slash sign with Ctrl + shortcut type "#" then replace tab keyword "/"

tags with no ending tags ~~use~~ use / sign at the end of the tag or before this arrow > such as  / >

In some cases, you have already built or written code that is not under or you don't want to use ~~too~~ too
for that use fragment

Also known as JSX fragments

It acts as a frame or all other that code to be written here

node-module is a dependency of your project if ~~it~~ mistakenly deleted this, then type npm install or npm i and you get that again dependency in your project

Don't write code into your public folder

App.js return type actually catched at index.html file ~~id~~ id="root"

you want to
Now use ~~js~~ bootstrap in alternative to CSS then type bootstrap on google and on introduction page you get CSS code with ~~link~~ href, copy that and paste it into your index.html file under public folder under head tag

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" ...">

Advanced ⁱⁿ Javascript

for loop based more concept which to array and object

① for each ()

② for in (that's not complete syntax) that just a keyword

③ for each () which is basically for array

array syntax
variable name here = [0, 1, 2, 3, 4];

for each syntax with array

variable name here. for each (here is inbuilt function);

that's basically function ()

In inbuilt function we use formal parameter such as x, y, z or abc or value, index

A function without name or here we don't need to create other function body or not call that too

x be automatically selected for demand, y for index

<script>

var x = [1, 2, 3, 4, 5];

x.forEach(function ()

formal parameters are x, y for getting output in that

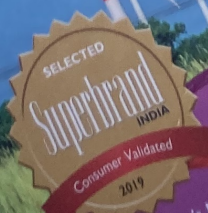
// write here what tasks to do

);

here it is

document.write(x + ":" + y + " (br>");

output element value 2 : 1



India's No. 1 Superbrand.

~~we can do for each by function body too~~
we can do for each by function body too

function abc (~~abc~~);
document.write (~~abc~~);

function body

② for in, generally used for object
object covers key/property and their value

Syntax

variable name here =

key/property here : value
name = "Rohi"

if you try to show object as output you'll get not nothing such as this text [object object]
for this use, if you want to see value of name key/property
x.name on document.write(x.name);
give you value of that key/property

IMPORTANT NOTES

for in syntax

for (let " " in " ")

in

standard keyword

write variable name here

variable name

here that must matches your object variable name

// write code here for what tasks to do

<script>

var variable name here = value

key/property: value
name: rohi
class: rohit

for (let " " in " ")

keyword

write variable name here for fully output

here that must matches your object variable name

This ~~is~~ automatically considered as []

document.write (y + " : " + x [y] + "
 ");

output is name: rohi
class: rohit

as output for key values only



</script>

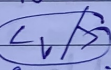


IMPORTANT NOTES


~~Variable name~~
~~for key~~
~~usually given a key value, this is~~
~~condition is variable name of taking output~~
~~must match~~
condition variable name for taking key values only
must matches the output taking variable
output taking variable automatically accepts as
key properly

React notation

-  using for file which is available at same location or path such as by vs code, set then i created a new folder as default path there, there was already App.js file, so for importing we use  `./` before file name notation

How to include any component in your project such as Navbar, you first to import then use tags 


these tags

 for including any component

✓ "write here what to include"
 } here it's Navbar or component name

✓ that's whole a component tag

o/c - shortcut used for code fast when you created a blank page then ~~o/c~~ by o/c you got already built template according to your file name in which you get import, export, return too, by default you got function based component such as export default function " "
 return (
 component name here ()
)

 this whole got by o/c

Create a module, module has extension of .mjs linking two modules or get ~~data~~ data to other module by export.

create a filename with name module1-mjs

~~var~~ x = "Rohi";
var y = "Rohit";

→ create a variable here and assigning things to that

Now

export

for export use export keyword the type default and type variable name here to which to export

export with default keyword its a way to export or type

→ export default "variable name here which to send";
here its

Now import this export to your other module filename

~~import~~ import this keyword
import "write any variable name such from" as x, y, z or q, b, c
this not interfere with output because its export type

→ here its

output will be shown on console.log()

→ write variable name here which we do import

module location here with . / notation for importing
here its ./module1-mjs.js

export type 2

```
var x = "Rohi";  
var y = "Rohit";
```

→ create a variables here and assigning string to that

Now export

for export use export keyword then here no need of default keyword and type variable name here which to export

all variables which to export must be included in that ~~code~~ braces such as

```
export {x};
```

In place of default keyword use curly braces

Now import which you already export

for import do

```
import { "here variable name must matches which to export before" } from 'single quotes'
```

module location here

with ./ notation for importing

output will be shown on console.log(" ");

write variable name here in which we do importing

```
import {x} from './module.mjs';
```


Import ^{and} export is generally a Javascript concept
Sending Information through props/property

```
function do "functionNamehere" () {  
  return (
```

```
  );  
}
```

for ^{the} use of props we ^{first} use this as a ~~parameter~~ parameter because maybe that function has any task to which we do change such as

```
function functionNamehere ( ) {  
  return (
```

// here is task Navbar ;

```
  );  
};
```

we want to change that Navbar to Home menu for that

• write ^{first} props as a parameter so that function

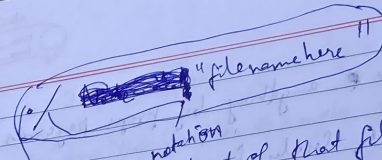
```
function functionNamehere (
```

here parameter is props)

```
  return
```

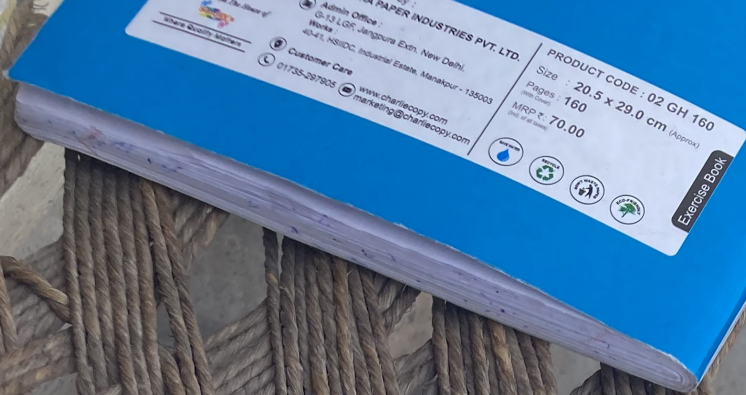
// here is task Navbar;

```
);
```


Node  "filenamehere"
 gives notation
 such as module for export
 module for import
 Node ./ "filenamehere"
 notation module.rings

When you creating any component first letter be in Capital such as ~~Navbar~~ Navbar.js
 for simplicity we first create components folder then write components ~~to~~ that

props provide you dynamic message
 modules also used for send information
 propTypes or props
~~propTypes~~ validation
 we first use to use shortcut and you get
 import propTypes from 'prop-types';
 Then Syntax
 " " propTypes =
 component name here ~~to apply this~~
 "genuine keyword here": propTypes.
 Keyword
 value type here such as string, number
 here its string



classmate
Date _____
Page _____

in the case of number

~~component name~~ ~~keyword~~
here \rightarrow keyword
"genuine-keyword" : propTypes, number,
}

in propTypes if you assigning number type value
to the genuine keyword then that value must be proper
curly braces { }

< Nav for "genuine keyword" = { } \rightarrow for number
O - here \rightarrow type value
write any number
here
component name

That means we declaring genuine keyword is of what
type ~~such~~ such as number, string

default props

If you mistakenly not assigned any value
to the genuine keyword then by default, default
value works

such as

"genuine keyword" here : { }
value here

}

Signature

component name here

number

curly braces

propTypes basic usage is to know that genuine keyword
assigned value of what type or we can set that

That means we do create a validation here such as when assigned a number to genuine keyword this will give you a error on console sent as failed prop type. In valid group 'little' of type 'string' supplied to (Number) expected 'number'. because we created validation for that after writing a whole component we do create a validation

validation is basically propTypes with default value must be of which type so that when get an console if we assigning wrong value

such as under component this first we will create

```
function Name() {  
  return
```

```
}  
export default
```

propTypes write in newline

condition is propTypes validation must be written under in component in which we using props
• after write whole that component

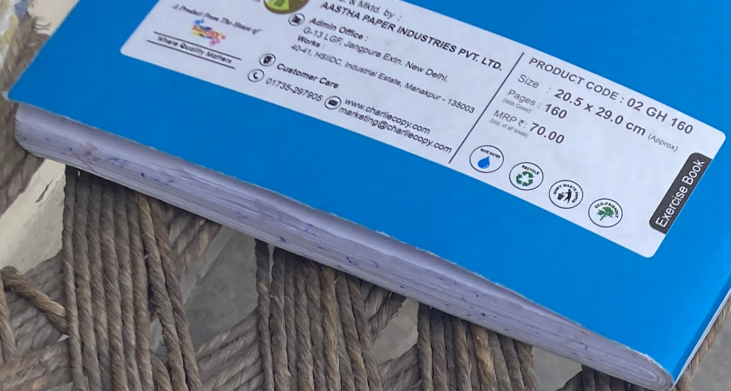
```
function Name() {
```

```
}  
export default
```

code of

```
import default function Name() {
```

```
}
```

classmate
Date _____
Page _____

export default
at the end
or this
will export
whole function
~~as whole component~~
as whole component
for use that into
other file

this will export
whole function such as
function abc() {
 return
}

Javascript in advance

Rest
if you have 6 elements, 3 be selected for
3 variables and left one are used for Rest
on console Rest give your left elements in a
array form

<script>
function abc() {
 document.write(x+y);
}
abc(2,3);

here are two variables generally
x, y, "rest operator"
with new variable "
...abc"

function body

call here

in most case add more actual parameter
(such as 5, 6) then formal parameter
left one elements you got to Rest as
array

for Rest to be working formal parameter has Rest syntax to
such as "...". Then here any variable name
3 dots
abc

on console, abc1 gives you left elements
 output is $\{a: 5, b: 6\}$ of rest on
 console(abc1);
 0: 5
 1: 6

Spread in object

How to update the object property value

<script>

```
var x = {  
  name: "Rohi",  
  class: "Rohit",
```

here its Rohit

// update by this $x.name = "sking here to whom update";$

now check on console

```
console.log(x);
```

output is

```
{name: 'Rohit', class: 'Rohit'}
```

value

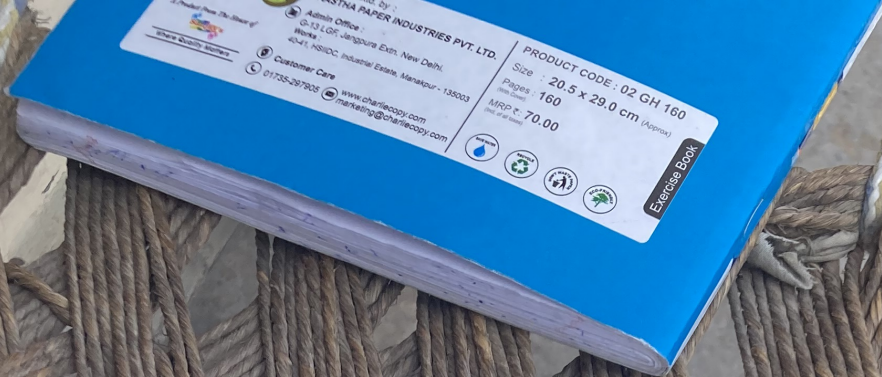
Now update any property of object by spread

Now create a new variable here

new variable // $var y = \{ \dots \}$ curly braces

this is
new
variable

• "variable name to which included
or based on to whom you
property value to be linked
for changed here its
x,"
spread
operator



classmate
Date _____
Page _____

"write here what to update in property value"
↓
here its class property and its value to
be update
such as class: "Rohit"

on console check this
console.log(y);
outputs {name: 'Rohit', class: 'Rohit'}

Rest in object

```
const x = {  
  name: "Rohit",  
  class: "Rohit",  
}
```

New way to show value or get a value
var { } = { }
↑ ↑
here enter property name to whom you get value
such as name
↑
here enter variable name from where you get
↑
on console log console.log({ })
↑
property name here to which you entered under curly braces
↑
such as here variable name is x; or object elements
↑
here its name curly braces
↑
output is Rohit

That was one ~~right~~ element, left on where goes
so we use here Rest

→ So we use this for using Rest
Pitavale

var (name) abc
 ↓
 post operator

on console
 console.log (abc);
 output is
 {class: Rohit}

Simple function

function body { function abc ()
 ↓
 call here // abc();

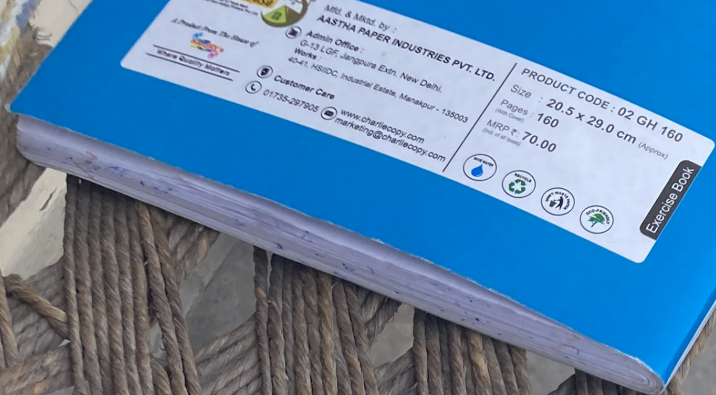
types function in modern way

~~function~~ var "function name here" = function ()
 ↓
 curly braces here its abc

// call here abc();

types var "function name here" = [] => arrow
 ↓
 curly braces here its abc
 // call here abc();

(That) generally called arrow function



classmate
Date _____
Page _____

if function has only one task to do on line line
then
simplified way of writing function is

var "function name here" = (^{square braces} _{array} ^{"here task to do"} _{console.log (^{any variable here})})

// call here abc();

more way to this
if you have only one parameter without square braces

var "function name here" = (^{parameter} _{formal parameter here} ^{"here task to do"} _{console.log (^{actual parameter here})})

// call here ~~abc~~ abc(_{here its 3})
actual parameter here
_{here its 3}

spread in array

var z = ["Rohi", "Rohit", "Rohit1", "Rohit2", "Rohit3"];

function body {
 function abc (_{formal parameters here such as x, y, z, a})
 console.log (_{document.write (x+y+z+a)})
}

variable name here which has entered at creating array how its z

Hook use `abc(z)`;

This way or `abc(z)` without spread outputs contains all ~~variables~~ and that's why `x`, `y`, `z` and a variables get undefined

To fix this we spread in array

in place of `abc(z)` use `abc(...z)`;

spread operator individually substituted as per the actual parameters formal parameters number of and in output undefined got to be not shown

such as `x` for 1st element
`y` for 2nd
`z` for 3rd
`4` for 4th

React

Hook means without creating a class use features
useState is type of Hook

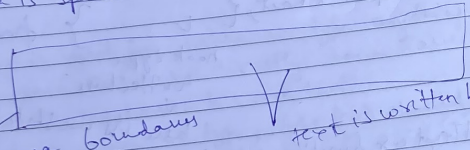


classmate
Date _____
Page _____

State means situation which right now, after type what situation is

Textarea is space where we write any text

that's means variable here takes value by default written hook



generally by bootstrap we get textarea snippet such as

```
<textarea class="form-control" id="exampleFormControlTextarea1" rows="3"></textarea>
```

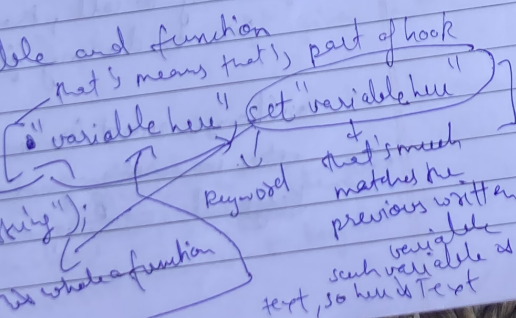
for this go google and search first bootstrap and in left side find form control, click that and now scroll down and get example tab, under this you get two div and you second one for textarea, use that without label tag or simply use textarea tag

This will show a textarea at output if you want to increase textarea height use rows="11" update that to value or number higher than 3, that's will add more rows to that

that's means variable here takes value by default written hook

Hook is based on variable and function

generally syntax var



here creating hook variable and function

hookname ("here its value")
here its value or default value

Keyword

that's means that's part of hook
that's means matches the previous written variable such variable as text, so here is Text

now, functions created and variables ho
^{not}
 before ~~hook~~ must import that hook type
 such as
 import React, { ^{hook type} } from 'react';
 ↓
 written here
 ↓
 here its use state

setText or function here to update or change anything
 that is our project based keyword that's means
 if we write or text anything under textarea that
 will do be shown on output and we will update
 this by setText

so keyword text comes as a value for textarea
 so now textarea tag will be
`<textarea class="form-control" id="exampleFormControlText
 rows="3" value={text}`
 area"

only braces here means that's text keyword
 is linked to hook or specifically
 linked to code written for creating variable
 or function or a hook

Direct functions don't need to create whole body, it's
 just only to call, that's available in React, it's
 for to do any task

onChange event ~~is a event~~ ~~that is used to~~ ~~create a function~~ ~~linked to that~~ ~~we are~~ ~~using~~
~~as parameter~~

such as

value function name = () =>

here its event keyword

onchange is used for when we type anything under empty space that's not directly shown after then we create mechanism to work that such as when we type, type text should be shown on output or under blank area, for that we type from any device that's not directly shown on output, we to update that to be seen on output

must has event parameter or keyword event

so function or setText is a part of onchange event do for onchange event to be work for that function linked to that event. target.value actually that's getting the value of text, that may be

~~the value of text~~ number or string or even character we used this getting value to update the text or that's text comes as parameter for function or setText value

setText(event, target.value) value got here

setText updating that getting value

Direct functions

such as toUpperCase() that's convert given text to the uppercase

toLowerCase() that's convert given text to the lowercase

we used event keyword here to forget the value that's means for onchange we first create function which has event parameter then use this same event keyword to forget value or get value

without that onchange not work

Textarea tag can work without class too so we can use them and textarea be shown on output

How to make Record of your text or each character

such as
Total character = 5 - on type 5 character in your textarea

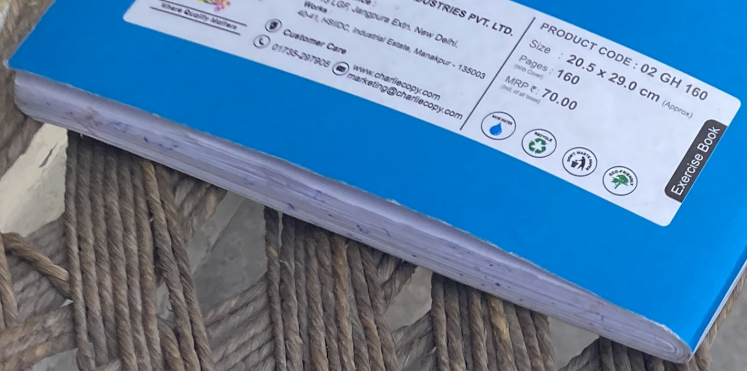
mechanism is use Total character = {text.length} ^{variable here}
length keyword ^{keyword} convert each character to count or in number

early for
browser for
we accept
here
available with
some property
~~keyword~~

mechanism is use Total words = {text.split(" ")}.
are space must be split function length so that

split("") works as space or the add space when we type space on your device
split("") add space or empty string to that
such as first write A , without split("")
function space not works
device, so when you give space between two character A B ^{space here on your device} will give you 2 words ^{keyword}
space and length

Preview is just copy of textarea which don't to edit, it's just to watch as textarea text be written, here be text written seen



classmate
Date _____
Page _____

~~if condition~~
~~if condition~~

dark mode if-else

if ("attribute here" == "color here")
strict equality better than

// update color here
object property value
set "attribute here" ==
color: "white";
backgroundColor: "black";

// update here button string
setButton ("string write here to
curly braces which to show on button");

else
set "attribute here" {
color: "black";
backgroundColor: "white";
}

this whole a
object property and value

setButton ("string write here to which to show on
button");
curly braces

generally
{
here write property name } = "here object"
to whom value
show at output
name to whom you
property
name linked"

in React
actually this is opposite
"here property name" = { "here object name to whom your property name is linked" }

state is used to do multiple works at one time
such as

- update text
- update object
- update button styling

- for update create one state
- for update text create one state too
- for update button styling create one state too

Not's means different update needs different states

Similarly for getting object into attri but
"attribute here" = { "object name here" }

value is also an attribute

How we create mechanism to ~~copy what state~~

~~one way~~ ~~direct function~~ ~~function~~ ~~sequence~~

~~returning value with return~~

~~if you want to return a value from a function, you have to use return statement. If you don't use return, the function will return undefined.~~

~~if you~~

~~if you want to return a value from a function, you have to use return statement. If you don't use return, the function will return undefined.~~

Javascript in advance

New way to call back function

<script>

var "function name here" = () => {

such as

abc

paywall for
setting the time
for daily
the task

// here task to do

}, "here time in ms);

as
such as 2000

for 25);

for daily task
after 2000s

use that same function here
which previously
used

So here, no need
of write again
function name

// function call here

abc

~~return~~

</script>

sg hard to say



if in case this structure you got such as
"functionname" ("functionname");
1 2

Part 1 means functionname1 calls then
functionname2

Direct functions also called task functions such as
functionname(); ^{Global scope}

↓
here function name also tells task
what task to do

That's means "function exists with inbuilt task"

Total character = "get here any number"

for that

mechanism is that

use { }

curly braces

~~for~~ to show number as
output

mechanism is "variable name" "property name"

↓
here its text

↓
here its length

Javascript is advance

~~basic~~ calling other function first even the
first function time is less

<script>

var abc = () => {

setTimeout(() => {


```
document.write(a);  
{, 2000};  
  
var abc = () => {  
  setTimeout(() => {  
    a = 90;  
    // callback call here "any variable here"();  
  }, 2000);  
}
```

In that, any variable to do as parameter to first function, that function acts as formal parameter, so, this formal parameter accepts whole first function as value to parameter such as

```
() => {  
  setTimeout(() => {  
    document.write(a);  
  }, 2000);  
}
```

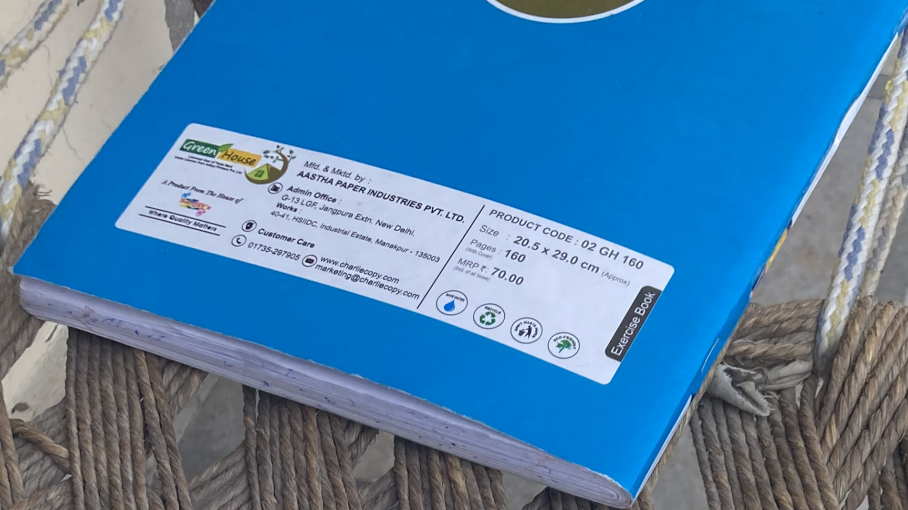
In that first function not works first even first function. `setTimeout` is less than second function. Now we have to call that function or `xc()` after done our necessary work of second function `xc()`; calling meaning first function calling here

```
<script>  
var abc = () => {  
  setTimeout(() => {  
    document.write(a);  
  }, 2000);  
}
```

whole function
not to
x

```
var abc = () => {  
  // any variable here  
  // here it's x
```

done



classmate
Date _____
Page _____

now use for call at time
set timeout () => <
a=90;
x();
for (3000);

abc1(abc);
ab1, first call
function
then abc function

asynchronous or async function or async()

async
async means if we first
go for first task
simultaneously, we for
second task, even third
task
here multiple works
done at same time

sync
sync means if we first one
task then go for second task
here one work at same time

~~await~~ await keyword
for reducing the overload due to asynchronous
, we use await keyword ~~for add~~ for add
sequence to your tasks such as this task must done
first then go for second task
for that use await keyword before that task

keyword
await "task1 here" // as this task successfully done then
await "task2 here" task2 goes to work

for using async function, you must create other function for use that such as

```
<script>
var "newfunctionname here" = async () => {
    here abc3
}
```

// now call this function here
abc();

Javascript in advance

Promise function

It's used to return something

we use call for
get first
function
then after

```
<script>
var "function-name here" = () => {
    // task here
}
```

```
var "newfunction-name here" = () => {
    // task here
}
```

and get() successful then we do use catch()

on successful call of set() then must call get() for not use ; then to dot set()

set().then().catch()

"write here which function to call"
↓
here its get(), so use here
function name
"write any new variable here for get error
such as error"



classmate
Date _____
Page _____

Now what we got error in error variable we have to view that to console for that we use arrow function such as

```
catch((err)) => { console.log("same variable here that must matches the previous one") }
```

curly braces

Promise function accepts two parameters here is error such as `except` or `reject`, or we say `resolve` or `reject`, `resolve` uses when not we get any error, `reject` to get any error

callback function with promise return

```
var abc = () => {  
  return
```

now we ~~don't~~ ~~use~~ ~~new~~ ~~keyword~~ before writing ~~promise~~ ~~function~~ which to return, that's necessary too so that promise function works

```
such as  
variable = () => {  
  return new Promise ( )
```

keyword
that has two parameters `resolve` or `reject`

```
(resolve, reject) => {
```

```
setTimeout ( ) => {  
  x = 90;
```

Now we working on mechanism that error get then get `reject`, otherwise `resolve` so in that case we previously or by default set that variable to be which has stored memory value

error full or none or error false, or error
 here
 for that
 // create variable here
 var "variable here" = "stored value here"
 such as
 error
 units false;

```

    if (!error) {
      resolve(); // resolve function
    }
    else {
      reject("write here what to show");
    }
  }, 2000)
})
  
```

Generally async function is used with Promise function

That's means we do first use Promise function in return, then we go for async function

Use Gmaps in your project in HTML
 first search google maps on google then click
 these links — at left side, then click share or
 embed map then click embed a map, copy that
 iframe tag link, paste that in your html file
 under body tag



classmate
Date _____
Page _____

Direct function with number of direct functions
~~where we use direct~~
we have to remove selected blue background
after text successfully selected

for selected we use select();

now we want to remove background so
we first get selected area for that we
getSelection() then now these selected background
by removeAllRanges() remove area or

such as
document.getSelection().removeAllRanges();

How to count that space is not considered as a
whole word
generally we do use "variable here".split(" ").length

~~we do use~~
But that's counts space as whole word
for we to work before length or total word count
now it's only way to overcome this to use new function
here called filter function and to apply condition
to that too

such as

filter(v) => { return "used that same variable here
which you used before
such as element
length !== 0" }
↓
"write any variable here
such as element"
↓
property ~~length~~

That's means we do condition here if any variable here
called element length not equals to zero then

return will replace ~~split function~~ ~~split~~ string by ~~condition~~
~~any variable~~ ~~split~~ ~~element~~ ~~length~~
 which has started element length

Basically filter function variable ~~replace~~
 split function string by

Disabled the button function by variable and
 property we disabled attribute

<button disabled = { "disabled here" }
 height text
 property
 height
 length

We apply condition to
 that that means

== 0

on no text length buttons should be disabled

Arrow function

Not whole arrow function
 Squared braces
 () => { }



Arrow function is of ~~two~~ types
① synchronous function ② asynchronous function

~~Asynchronous function works with async function~~
Promise function, setTimeout function, not
works without async function or it's part
output not shows

Arrow function is used to create one function
under other function
such as

```
var abc = ( ) => {  
  setTimeout(() => {  
    // "such as in this"  
  })  
}
```

that's a
arrow
function too

for your project easily be searchable by the
Search Engine Optimization
go under index.html file and go to content
under head tag and write keywords under
" ", such as Text converter, convert text
uppercase to lower / lowercase to uppercase
not's or

Function Based Components

- ① Stateless
- ② New Props are directly used
such as { props: "keyword" }
- ③ After updation these able to
use usestate hook

Class Based Components

- ① Stateful

Props are immutable or not changeable
such as

- < "component name here" "keyword" = "its value here" />
- you then again assign value to that same keyword
- < "component name here" "keyword" = "its new value" />

That will broke the output, nothing will display a output

Class based components syntax

First we do import React from 'react';
Syntax here class "classname here" extends React.Component

render() {
 // defines function that runs in class before return
 return (
 JSX Element here
);
}

this is mainly class name

Curly braces

</>
/>
}

Props in class components

in function we used here to use
< this.props "keyword" />



classmate
Date _____
Page _____

Props with ~~the~~ children
for that component included must be
turn into ~~the~~ tag such as
<Class /> - A whole component tag
that will turn into

<Class> </Class>

Now create children to this tag

<Class "Keyword here" = "value here">

// children write here

<h1> Hello </h1>

<h2> Hello! </h2>

</Class> ~~keyword here~~

we don't need to write again keyword

so that means that children linked to that keyword
so we first create props for that keyword then
for that children

such as

<h1> Rohit < props. "keyword here" > </h1>
< props. children > </h1>